

Přidělování IPv6 adres v prostředí GNS3

IPv6 Address Assignment in GNS3 Tool

Jakub Dzian

Bakalářská práce

Vedoucí práce: Ing. Pavel Nevlud

Ostrava, 2021

Abstrakt

Cílem bakalářské práce je popis možností získání IP adresy v prostředí protokolu IPv6. Teoretická část popisuje způsoby adresování zařízení a získání jiných konfiguračních parametrů, jako je DNS server, prostřednictvím technologií SLAAC, zero configuration, stateless DHCPv6 a statefull DHCPv6. V praktické části je popsáno jakým způsobem se pracuje v prostředí GNS3 souběžně s virtuálními zařízeními s operačním systémem Ubuntu. V této části je také popsána konfigurace malé místní sítě skládající se z DHCPv6 serveru a 3 klientů, kde každý z nich bude mít přiřazenou IPv6 adresu jiným způsobem. Následuje ověření konfigurace pomocí screenshotů a zhodnocení popsaných metod.

Klíčová slova

DHCPv6; GNS3; IPv6; ISC DHCP; Linux; RADVD; SLAAC; Ubuntu; zeroconf;

Abstract

Purpose of my bachelors thesis is to describe options of obtaining IP address in the IPv6 environment. Theoretical part describes different ways of device addressing and gaining other configuration parameters, for example DNS server, through technology called SLAAC, zero configuration, stateless and statefull DHCPv6. In the practical part is described how to work in GNS3 environment in parallel with virtual machines running on operation system Ubuntu. In this part there is also described configuration of LAN consisting of DHCPv6 server and 3 clients, where each of them will have assigned address by different method. Verification by screenshots and evaluation of described methods follows.

Keywords

DHCPv6; GNS3; IPv6; ISC DHCP; Linux; RADVD; SLAAC; Ubuntu; zeroconf;

Poděkování

Tímto bych chtěl poděkovat mému vedoucímu práce Ing. Pavlu Nevludovi za čas vyhrazený pro konzultace, praktické rady k řešení a konstruktivní výtky k teoretickému rozboru. Také bych rád poděkoval svému bratrovi za korekturu a ostatním členům rodiny za trpělivost.

Obsah

Seznam použitých symbolů a zkratek	6
Seznam obrázků	8
Seznam tabulek	10
1 Úvod	11
2 Popis IPv6 adres a způsoby přidělování	13
2.1 IP adresa	13
2.2 IPv4 adresa	13
2.3 IPv6 adresa	13
2.4 Druhy IPv6 adres	14
2.5 Multicastové adresy	15
2.6 Formát globální jedinečné IPv6 adresy	15
2.7 Interface Identifier	16
2.8 Manuální konfigurace	17
2.9 Zero Configuration	17
2.10 Stateless Address Autoconfiguration	20
2.11 DUID	23
2.12 Stateless DHCPv6	25
2.13 Statefull DHCPv6	26
3 Popis grafického síťového simulátoru GNS3	29
3.1 Instalace a příprava GNS3 prostředí	29
3.2 Grafické prostředí	30
3.3 Virtual Machine	31
3.4 Instalace Ubuntu ve VirtualBoxu	32

4	Návrh zapojení pro testování automatického přidělování IPv6 adres	34
4.1	Konfigurace Ubuntu sloužící jako server	35
4.2	Nastavení stateless Ubuntu klienta	38
4.3	Nastavení statefull Ubuntu klienta	39
4.4	Nastavení statefull fixního Ubuntu klienta	39
4.5	Ověření navrženého řešení	40
5	Závěr	47
	Literatura	49
	Přílohy	52

Seznam použitých zkratek a symbolů

CCNA	– Cisco Certified Network Associate
CCNP	– Cisco Certified Network Professional
CPT	– Cisco Packet Tracer
DAD	– Duplicate Address Detection
DHCP	– Dynamic Host Configuration Protocol
DHCPv6	– Dynamic Host Configuration Protocol version 6
DNS SD	– Domain Name System Service Discovery
DNS	– Domain Name System
DUID	– DHCP Unique Identifier
DUID-EN	– DUID based on Enterprise Number
DUID-LL	– DUID based on link-layer address
DUID-LLT	– DUID based on link-local address plus time
DUID-UUID	– DUID based on Universaly Unique Identifier
EUI-64	– Extended Unique Identifier 64-bit
GNS3	– Graphical Network Simulator-3
IA_NA	– Identity Association for Non-temporary Address
IA_PD	– Identity Association for Prefix Delegation
IA_TA	– Identity Association for Temporary Address
IA	– Identity Association
IA	– Identity Association
IAID	– International Academy for Intercultural Development
IANA	– Internet Assigned Numbers Authority
ICMPv6	– Internet Control Message Protocol version 6
ID	– Identifier
IDE	– Integrated Development Environment
IID	– Interface Identifier
IOS	– Internetwork Operating System
IP	– Internet Protocol

IPv4	– Internet Protocol version 4
IPv6	– Internet Protocol version 6
ISC	– Internet Systems Consortium
ISP	– Internet Service Provider
LAN	– Local Area Network
LIR	– Local Internet Registry
LTS	– Long-Term Service
MAC	– Media Access Control
MD5	– Message Digest 5 Algorithm
mDNS	– Multicast Domain Name System
MSB	– Most Significant Bit
ND	– Neighbour Discovery
NIC	– Network Interface Controller
NTP	– Network Time Protocol
OSI	– Open Systems Interconnection
OUI	– Organizationally Unique Identifier
PC	– Personal Computer
RA	– Router Advertisement
RADVD	– Router Advertisement Daemon
RAM	– Random Access Memory
RFC	– Request For Comments
RIR	– Regional Internet Registry
RS	– Router Solicitation
SIP	– Session Initiation Protocol
SLAAC	– Stateless Address Autoconfiguration
TCP	– Transmission Control Protocol
UDP	– User Datagram Protocol
UTC	– Coordinated Universal Time
UTF-8	– Unicode Transformation Forat - 8 bit
UUID	– Universaly Unique Identifier

Seznam obrázků

2.1	Formát globální jedinečné IPv6 adresy, převzato z [9]	16
2.2	Objevování služeb v lokální síti	19
2.3	Ukázka obecné mDNS-SD komunikace	20
2.4	Formát Router Solicitation zprávy, převzato z [19]	21
2.5	Formát Router Advertisement zprávy, převzato z [19]	21
2.6	Sekce Prefix Informations obsažená v RA zprávě, převzato z [19]	22
2.7	Typy DUID, převzato z [8]	23
2.8	Formát DUID-LLT, převzato z [8]	24
2.9	Formát DUID-EN, převzato z [8]	24
2.10	Formát DUID-UUID, převzato z [8]	25
2.11	Průběh DHCPv6 operací v závislosti na relay-agentovi	26
2.12	Formát IA_NA, převzato z [8]	27
2.13	Průběh stavových DHCPv6 operací v závislosti na relay-agentovi	28
3.1	GNS3 grafické prostředí	31
3.2	GNS3 VM ve VirtualBoxu	32
3.3	GNS3 webové prostředí v prohlížeči	32
3.4	Nainstované Ubuntu VM ve VirtualBoxu	33
4.1	Topologie v prostřední GNS3	35
4.2	Databáze leases serveru	40
4.3	Databáze leases serveru	40
4.4	Výpis IP adresy serveru	41
4.5	Databáze leases serveru	41
4.6	Výpis IP adres, DNS serverů a doménových jmen stateless klienta	42
4.7	Databáze leases stateless klienta	42
4.8	Výpis ip address, DNS serverů a doménových jmen statefull klienta	43
4.9	Databáze leases statefull klienta	43
4.10	Výpis ip address, DNS serverů a doménových jmen statefull klienta s fixní adresou	44

4.11	Databáze leases statefull klienta s fixní adresou	44
4.12	Databáze leases statefull klienta s fixní adresou	45
4.13	Zachycení DUID klienta pomocí tcpdump, zobrazené ve Wiresharku	45
4.14	Stateless komunikace DHCPv6 serveru s 1. klientem	46
4.15	Statefull komunikace DHCPv6 serveru s 2. klientem	46
1	Nefiltrovaná komunikace 2. klienta se serverem	62

Seznam tabulek

2.1	Kombinace flagů obsažených v Router Advertisementu a jejich význam	22
-----	--	----

Kapitola 1

Úvod

V dnešní uspěchané a pohodlné době se dbá na rychlost a jednoduchost, proto se v síťových technologiích přiklání k přístupu "Plug and Play", který symbolizuje to, že není zapotřebí žádné konfigurace zařízení, což u IP adres představuje technologie DHCP. DHCP neodmyslitelně patří k našemu každodennímu fungování, ať už se doma nebo v kavárně připojíte k Wi-Fi, bude vám automaticky přidělena adresa a s ní přístup k internetu. V dnešní době existuje nespočet zařízení s přístupem k internetu a každé z nich potřebuje svou adresu, proto dochází k vyčerpání IPv4 adresního prostoru. S vývojem technologií se pomalu, ale jistě přechází k IPv6, která vyřešila některé z chyb a nedostatků zastaralé IPv4, která vznikla v 70. letech. Prostředí GNS3 je velmi užitečné, zejména v situacích když nemáte přístup k specializované síťové laboratoři. Můžete si do něj nahrát systém reálného zařízení, vytvořit si malou či velkou místní síť a nasimulovat síťový provoz. Pomocí GNS3 lze otestovat konfiguraci části sítě, čímž lze zamezit neočekávanému chování, či výpadku sítě, proto v GNS3 vidím velký potenciál.

Cílem mé práce je, aby čtenář porozuměl problematice přidělování adres v protokolu IPv6. Práce je rozdělená do několika kapitol. V první kapitole je teoretický popis IPv6 adres, jejich typů a poté způsoby jejich přidělení, kde jsem kladl důraz na průběh komunikace a jejich chování. Ve druhé kapitole je popsáno GNS3 prostředí a jak jej nainstalovat a zprovoznit ke správnému fungování. Dále také uvádí jak zprovoznit zařízení běžící na operačním systému Ubuntu ve virtualizačním prostředí Oracle VM VirtualBox. Třetí kapitola je věnována praktické části, kde se nachází popis postupu při konfiguraci Ubuntu zařízení v GNS3 prostředí. V rámci GNS3 prostředí budeme pracovat se switchem a se 4 Ubuntu desktope. Jedno ze zařízení běžící na operačním systému Ubuntu slouží jako DHCPv6 server a zbylé 3, které představují klienty, kterým bude různými způsoby přidělena IPv6 adresa a další konfigurační parametry. DHCPv6 služby na serveru budou zprostředkovány pomocí daemonů ISC DHCP server a RADVD. První klient bude mít přidělenou IPv6 adresu pomocí SLAACu a DNS server díky stateless serveru. Zbývající dva klienti budou mít přidělenou adresu od statefull DHCPv6. První z nich bude mít přidělenou adresu z adresního prostoru tzv. poolu a druhý dostane fixní IPv6 adresu mimo adresní prostor. Server klienta rozezná na základě

client-id. Popis konfigurace může sloužit jako příručka, nebo návod. Součástí této kapitoly je také ukázka ověření fungování zapojení a konfigurace. V příloze práce jsou důležité konfigurační soubory.

Kapitola 2

Popis IPv6 adres a způsoby přidělování

2.1 IP adresa

IP adresa je číselný, jednoznačný identifikátor přidělený každému zařízení připojenému do počítačové sítě, která používá internetový protokol ke komunikaci. IP adresy se používají na routerech neboli rozdělovačích, které pracují na 3. vrstvě OSI referenčního modelu.

2.2 IPv4 adresa

IP adresa verze 4 je definována 32 bitovým číslem. IPv4 adresa je rozdělena do 4 částí, každou část tvoří 8 bitů, což jsou čísla 0 až 255 v desítkové soustavě. Každá část IP adresy je oddělená tečkou. Počet IP adres verze 4 je 2^{32} , což je v přepočtu $4 \cdot 10^9$ (4 miliardy).

- např.: IPv4 adresa v desítkové soustavě (běžný zápis): 192.168.1.1
- výše uvedená IPv4 adresa zapsaná ve dvojkové soustavě (bitech):
11000000.10101000.00000001.00000001
- IPv4 adresa DNS serveru Googlu: 8.8.8.8

2.3 IPv6 adresa

IP adresa verze 6 byla vytvořena z důvodu nedostatku IPv4 adres. Tvůrci se obávali nedostatku adres v budoucnosti natolik, že délka IPv6 adres je čtyřnásobná 2^{128} , což je v přepočtu $3,4 \cdot 10^{38}$ adres. IPv6 adresa je definována 128 bitovým číslem. IPv6 adresa se standardně zapisuje po osmi skupinách, které jsou odděleny pomocí dvojteček. Jednotlivé skupiny jsou tvořeny 16 bity, které vyjadřují hexadecimální čísla [1].

- $x:x:x:x:x:x:x$, kde x jsou 1 až 4 hexadecimální čísla

- např.: 2001:4860:4860:0000:0000:0000:8888

Jak lze vidět, takový zápis není příliš přehledný, ani úsporný, proto se využívá kanonický zápis. Kanonický zápis se snaží o zkrácení zápisu pomocí několika pravidel podle RFC 5952 [2].

- Hexadecimální čísla A až F se píšou malými znaky. Toto pravidlo platí z důvodu, aby se zamezilo lidské chybě v podobě záměnění B s 8, nebo D s 0 a podobně.
- Vynechávání 0 zleva v jednotlivých skupinách je povinné. Musí se však dávat pozor na to, že zkracovat lze pouze zleva, jinak by mohlo dojít ke zkrácení z "4860" na "486", což by bylo chybné a ve výsledku by to znamenalo "0486".
- Lze vynechat skupiny skládající se pouze z 0 pomocí dvou dvouteček za sebou "::". Takovéto zkrácení se používá pouze jednou a v nejdelší řadě nulových skupin. Pokud jsou dvě stejně dlouhé řady nul, tak se pravidlo aplikuje na první z nich. Zkrácení musí pojmut všechny nuly v řadě a nesmí tak vzniknout chybný tvar jako je "0::", nebo "::0".

Pomocí výše zmíněných pravidel tedy lze zkrátit zápis IPv6 adresy 2 následujícími způsoby:

- V prvním příkladu lze vidět vynechávání nul v jednotlivých skupinách. IPv6 adresu: 2001:0abc:00de:000f:1234:0567:0089:a000, kterou zapíšeme ve zkráceném tvaru: 2001:abc:de:f:1234:567:89:a000
- Druhý příklad je proveden na výše již uvedené adrese DNS serveru Googlu: 2001:4860:4860:0000:0000:0000:8888, jejíž správný zápis vypadá následovně: 2001:4860:4860::8888

Pokud by uživatel neměl přístup k DNS serveru a chtěl by se dostat na webové stránky, IPv6 adresa se v URL zapisuje do hranatých závorek, protože dvojtečky se běžně používají k oddělení portu od jména nebo adresy [3][4].

- např.: DNS adresa serveru google.com se zapisuje: http://[2001:4860:4860::8888]

2.4 Druhy IPv6 adres

Existují 3 druhy IPv6

- Unicast - Identifikuje jedno síťové rozhraní stejně jako u předchůdce, IPv4.
- Multicast - Je skupinové adresování, kde odeslané data musí být doručeny všem členům skupiny (např. všem DHCPv6 serverům)
- Anycast - Nazývá se výběrové adresování a označuje určitou skupinu stejně jako u multicastu, ale rozdíl je v tom, že data však dorazí pouze jednomu členu skupiny a to tomu nejbližšímu.

Jak si mnozí mohli povšimnout, tak narozdíl od IPv4, zde chybí *broadcast*, který rozesílá zprávy všem. V IPv6 byl vynechán z důvodu častého vzniku tzv. "broadcastových bouří".

Zde jsou uvedeny některé IPv6 adresy z IANA registru se speciálním významem [5][6][7]:

- ::1/128 - Loopback
- fe80::/10 - Link-Local Unicast
- fc00::/7 - Unique-Local
- ff00::/8 - Multicast
- 2000::/3 - Global Unicast

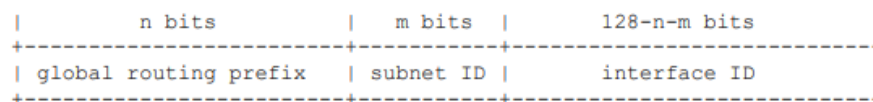
2.5 Multicastové adresy

Zde uvedu dvě multicastové adresy, které se využívají při SLAACu, či DHCPv6 a budou se dále objevovat [8].

- **All DHCP Relay Agents and Servers (ff02::1:2)** - Tato multicastová adresa je označována, jako link-scope, což znamená, že její dosah je pouze linka. Tuto adresu využívá klient ke komunikaci se sousedícími relay-agenty a nebo servery. Klient využívá UDP port 546 k tzv. poslouchání příchozích zpráv.
- **All DHCP Servers (ff05::1:3)** - Tato multicastová adresa má také omezený místní dosah a její označení je site scope. Tuto adresu využívá relay-agent ke komunikaci se serverem. Používá ji ze dvou důvodů, prvním je, že chce poslat zprávu všem DHCP serverům. Druhý důvod je neznalost unicastové adresy serveru. Servery a relay-agenti využívají k poslouchání UDP port 547.

2.6 Formát globální jedinečné IPv6 adresy

Globální adresy jsou přidělovány internetovými poskytovateli (ISP) na základě prefixů přidělenými LIR (Local Internet Registry), podle pravidel RIR (Regional Internet Registry). Na základě přiděleného prefixu svým zákazníkům přidělí delší prefixy, na jejichž základě lze zvenčí identifikovat celou síť poskytovatele pomocí záznamu ve směrovacích tabulkách. Struktura IPv6 adresy se skládá ze 3 částí stejně jako IPv4 adresa. Tyto části se nazývají: globální směrovací prefix (global routing prefix), identifikátor podsítě (subnet ID) a identifikátor rozhraní (interface ID). Délky jednotlivých částí jsou definovány pouze obecně v RFC 3587 [9], ale podle aktuálních pravidel pro přidělování se používá délka globálního směrovacího prefixu 48 bitů a identifikátor podsítě 16 bitů. Pouze identifikátor rozhraní má konstantní délku 64 bitů.



Obrázek 2.1: Formát globální jedinečné IPv6 adresy, převzato z [9]

Globální směrovací prefix - Tato část bývá označována jako veřejná topologie. Je přidělen LIR a identifikuje koncové sítě. Některé z menších koncových sítí mohou mít délku prefixu až 64 bitů.

Identifikátor podsítě - Správce sítě využívá tuto část k rozlišení jednotlivých podsítí ve své síti. Označuje tzv. místní topologii. Délka identifikátoru podsítě je závislá na délce globálního směrovacího prefixu, protože tyto části mají dohromady vymezených 64 bitů. Proto běžná délka bývá 16 nebo 8 bitů, ale pokud není potřeba rozdělení sítí, není potřeba ani identifikátoru podsítě. V takovém případě má identifikátor podsítě velikost 0 bitů [4].

2.7 Interface Identifier

Identifikátor rozhraní, poslední část globální jedinečné IPv6 adresy, označovaný jako IID se používá k tvorbě unicastových adres připojením za prefix dané sítě. IID má velikost 64 bitů. IID lze vytvořit 3 různými způsoby:

- Manuálně
- Modifikované EUI-64
- Náhodně vygenerované

Manuálně vytvořené IID - Je manuálně nakonfigurováno uživatelem. Je to časově náročné a je potřeba mít přehled o již přidělených IID. Využití může být například u adres síťové infrastruktury.

Modifikované EUI-64 (Extended Unique Identifier) - Je vytvořeno pomocí úpravy MAC adresy, která má však pouze 48 bitů. Ukázka procesu tvorby modifikovaného EUI-64:

1. 48 bitová MAC adresa hosta je např.: 00-12-34-56-78-9a.
2. MAC adresu rozdělíme v polovině na 2 části po 24 bitech. První část se nazývá OUI (Organizationally Unique Identifier), která identifikuje výrobce. Druhá část se nazývá NIC (Network Interface Controller). Máme tedy OUI: 00-12-34 a NIC: 56-78-9a.
3. Doprostřed vložíme hexadecimální hodnotu "ffe". Výsledek vypadá následovně: 00-12-34-ff-fe-56-78-9a.
4. Posledním krokem je obrácení 7. bitu zleva. Převédeme si tedy první 2 hexadecimální čísla na binární: 0000 0000, invertujeme 7. bit zleva: 0000 0010. Po převedení zpět na hexadecimální číslo (02) konečný tvar modifikovaného EUI-64 je: 02-12-34-ff-fe-56-78-9a [1] [10].

Náhodně vygenerované - Používá se například u zařízení běžících na Windows Vista, nebo Windows Server 2008 [11]. Je zapotřebí, aby operační systém vygeneroval vždy stejnou hodnotu při spuštění, proto si ukládá počet iterací, které byly potřebné k vygenerování jedinečného náhodného identifikátoru pomocí hashovací funkce MD5. Proces je podrobně popsán v RFC4941 [12] [13].

2.8 Manuální konfigurace

Nejjednodušším typem konfigurace je manuální. Nejjednodušší je z pohledu na počet akcí potřebných ke konfiguraci. Naopak je zapotřebí podrobná znalost, správce těchto zařízení musí provést konfiguraci správně podle standardu. Tento typ konfigurace je nepraktický, ačkoliv je potřebný u některých zařízení, například routerů.

Parametry, které je zapotřebí nakonfigurovat:

- IPv6 adresa, síťová maska

konfigurace v linuxu: `sudo ip -6 address add adresa/maska dev rozhraní`

např.: `sudo ip -6 address add 2001:db8:cafe:1::1/64 dev enp0s3`

- Směrování

konfigurace v linuxu: `sudo ip -6 route add adresa dev rozhraní`

např.: `sudo ip -6 route add 2001:db8:cafe:1::10 dev enp0s3`

- IPv6 adresa DNS serveru

konfigurace v linuxu: `sudo nano /etc/resolv.conf`, kde se doplní, nebo přepíše řádek:

`nameserver x.x.x.x`, např. `na: nameserver 2001:4860:4860::8888`

2.9 Zero Configuration

Používá se také zkrácené označení zeroconf. Zeroconf umožňuje automatické vytvoření počítačové sítě. Požadavky jsou propojení počítačů a síťových periférií. Použitelná síť se vytváří na základě souboru internetového protokolu (Internet Protocol Suite), běžně známého jako TCP/IP. Z technického pohledu je zeroconf kombinace tří hlavních technologií: automatické přiřazení síťových adres zařízením, automatická distribuce a rozlišení počítačových jmen (hostnames) a automatické zjišťování umístění síťových zařízení. Tím dokáže nahradit centrální služby jako DHCP, DNS a ND. Zeroconf nevyžaduje zásahy uživatele pro inicializaci a řízení, tento systém si například vytváří mapu sítí a adres (Directory Service) a uděluje automaticky jména zařízení na základě výrobce zařízení, jako je značka a číslo zařízení. Jedno z prvních využití této technologie v 80. letech pro LAN systémy byl AppleTalk v začátcích Macintosh computers [14]. Projekt Avahi umožňuje implementaci zeroconfu

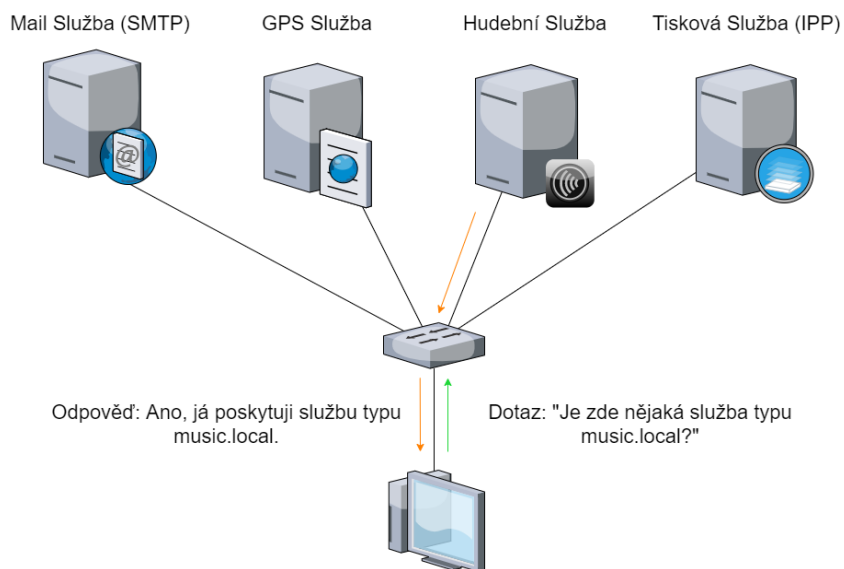
na operačním systému Linux, zároveň Avahi implementuje binární knihovny, které umí emulovat Bonjour, což je implementace zeroconfu od společnosti Apple pro operační systémy MacOS a iOS.

Generování link-local IPv6 adres probíhá na základě dosazování EUI-64 vytvořeného na základě MAC adresy za prefix sítě. Prefix sítě může mít až 64 bitů, začíná fe80:: a maska tohoto prefixu je /10. Zařízení je zodpovědné za výběr a jedinečnost adresy, proto je kontrolována pomocí Duplicate Address Detection (DAD), který je podrobněji popsán v další sekci o SLAAC. Link-local adresa je velmi důležitá ke komunikaci zařízení bez klávesnice a monitoru, jako je například tiskárna.

Multicast DNS zkráceně mDNS. Lidé raději využívají jména a ne adresy, proto existuje DNS. IP adresy jsou složitější na zapamatování a mohou se měnit, narozdíl od jmen domén, které ve většině případů zůstávají nezměněné. DNS má velkou výhodu pro mobilní zařízení, které se často připojují k různým sítím, s čímž se mění i IP adresa zařízení, proto je lepší mít stále jméno. mDNS se nesnaží soutěžit s běžnou DNS, ale slouží jako pojistka, pokud by DNS server nebyl k dispozici. mDNS poskytuje možnost tvorby doménových záznamů a odkazovat se na jednotlivá zařízení v LAN, díky jejich jmen. Host vytváří mDNS žádosti k přeložení jména na IPv6 adresu. Jak už z názvu vypovídá, tak k lokálním (jména končící na: .local) žádostem se využívá multicastová adresa ff02::fb a přiřazený port s číslem 5353. Klientovi se vrátí UDP odpověď obsahující požadovanou IPv6 adresu na základě doménového jména. Pokud by však žádná odpověď nedorazila, žádost je přeposlána (retransmitted) několikrát. mDNS používá UTF-8 k zakódování názvů záznamů o prostředku. Aby se zamezilo tomu, že host bude požadovat překlad jedné domény vícekrát, existuje proto známý seznam odpovědí (Known Answer List), který si ukládá každý úspěšný dotaz. Tímto způsobem je možné šetřit šířku pásma (bandwidth). Služba mDNS je podrobněji popsána v RFC 6762 [15]. K využití mDNS je však zapotřebí znát lokální názvy jednotlivých stanic a je možné, že člověk udělá chybu nebo zapomene doménové jméno. Z tohoto důvodu mDNS využívá technologii, která umožňuje vyhledávat zařízení a známé služby, touto technologií je DNS Service Discovery.

DNS Service Discovery - DNS SD je mechanismus, který umožňuje zeroconf objevit, které služby jsou právě dostupné v síti, bez předešlé znalosti jmen. Typickým příkladem může být například tisk. Klient pošle žádost do LAN, ve které požaduje IP adresu libovolné dostupné tiskárny. Tato technologie objevování opět využívá multicastových adres s prefixem ff00::/8. Veškeré stanice se připojí do této skupiny a naslouchají [16].

Objevování Služeb



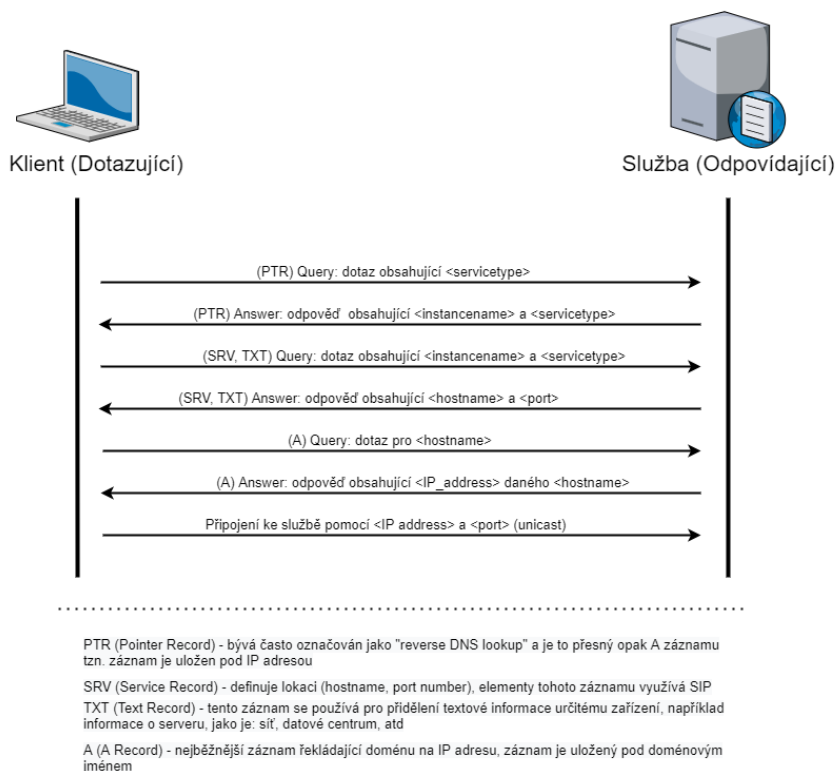
Obrázek 2.2: Objevování služeb v lokální síti

Proces vyhledávání služeb:

- Právě připojená zařízení vyšle informační zprávu, ve které uvádí informaci o připojení a své parametry.
- Zařízení pošle žádost na konkrétní multicast a port, ve které uvede požadované služby.
- Veškerá zařízení dané multicastové skupiny obdrží tuto žádost, pokud podmínkám nevyhovují, ale znají zařízení poskytující tuto službu, tak mu předají tuto informaci. Pokud však zařízení přímo odpovídá požadavkům, tak odešle nabídku.
- Zařízení, které odeslalo žádost počká na odpovědi a vybere si tak jednoho z poskytovatelů dané služby.
- Zařízení před opuštěním musí odeslat zprávu informující o této události.

Je důležité ještě zdůraznit, že se nejedná o objevování zařízení, ale o objevování služeb [17] [18].

mDNS-SD komunikace



Obrázek 2.3: Ukázka obecné mDNS-SD komunikace

2.10 Stateless Address Autoconfiguration

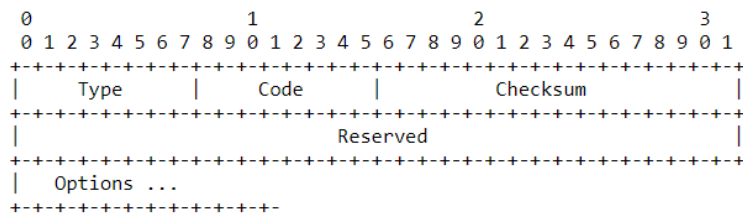
SLAAC je bezstavová automatická konfigurace adres. SLAAC je metoda, díky které si klient vytvoří unikátní globální IPv6 adresu bez předchozí konfigurace, nebo DHCPv6 serveru. SLAAC u této metody využívá víceúčelový protokol pro ohlašování chyb při přenosu paketů a testování dosažitelnosti ICMPv6 a klient získává prefix k vytvoření globální unicastové adresy pomocí směrovacích žádostí RS a směrovacích ohlášení RA.

Proces získávání globální unicastové adresy po zapnutí zařízení, nebo rozhraní:

1. Nody (host i router) si vygenerují link-local adresu připojením identifikátoru rozhraní (IID) k link-local prefixu (fe80::). Tvorba IID je podrobně popsána v sekci 2.7.
2. Před přiřazením, nebo použitím právě vytvořené link-local adresy je zapotřebí si ověřit její jedinečnost (zda ji nepoužívá jiné zařízení na lince) pomocí Duplicate Address Detection (DAD). K ověření se využívá ICMPv6 Neighbor Discovery, který je podrobněji popsán v doporučení RFC4861 [19]. Ověření probíhá, tak že host musí vstoupit do all-nodes multicast skupiny

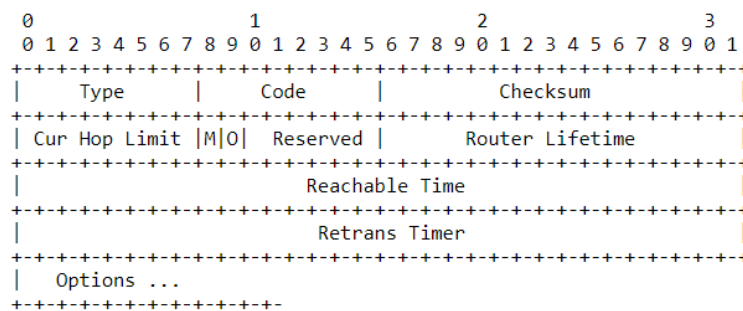
(ff02::1:) a odešle ICMPv6 Neighbor Solicitation zprávu, kde jako cílovou adresu vyplní tzv. orientační (tentative) adresu, kterou si vytvořil výše uvedeným způsobem. Pokud se na lince nachází zařízení s takovou adresou, tak odešle Neighbor Advertisement, který dává na vědomí, že takovou adresu vlastní. To pro hosta znamená, že je zapotřebí, aby přepsal defaultní IID nově vygenerovaným pomocí operačního systému. A tento proces probíhá dokud žádné zařízení neodpoví na Neighbor Solicitation pomocí Neighbor Advertisement. Jakmile je ověřena jedinečnost link-local adresy, tak je možno ji přidělit danému rozhraní. To umožní spojení sousedských (neighbor) nodů na IP-úrovni. Link-local adresy mají nekonečný (infinite) lifetime, to znamená, že platnost této adresy nikdy nevyprší, pokud nedojde k rozpojení linky.

- Host si může počkat na pravidelný Router Advertisement, ale pokud by dlouho nepřicházel, tak si host o něj požádá pomocí Router Solicitation, kde cílovou adresou je multicastová skupina všech routerů (ff02::2:).



Obrázek 2.4: Formát Router Solicitation zprávy, převzato z [19]

- Router Advertisement specifikuje jakou autokonfiguraci by host měl provést. Záleží na M a O bitech RA zprávy, viz obrázek 2.5.



Obrázek 2.5: Formát Router Advertisement zprávy, převzato z [19]

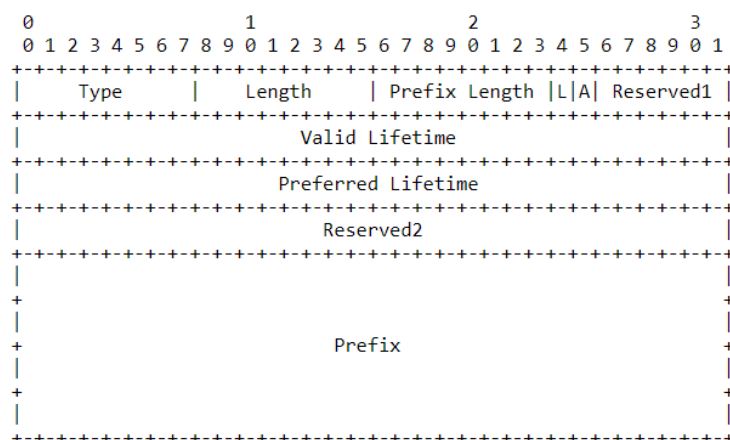
M označuje Managed address configuration flag a O označuje Other configuration flag. Oba flagy mohou nabývat hodnoty 0, nebo 1. V tabulce uvedené níže jsou uvedeny jednotlivé kombinace flagů a co znamenají pro hosta z hlediska autokonfigurace a DHCPv6.

Flag		Význam
M (managed)	O (other)	
1	0	Statefull DHCPv6
0	1	SLAAC + Stateless DHCPv6
0	0	DHCPv6 není k dispozici

Tabulka 2.1: Kombinace flagů obsažených v Router Advertisementu a jejich význam

Pokud se však v blízkosti nenachází žádný router, od kterého by host přijal Router Advertisement, tak zahájí autokonfiguraci pomocí statefull DHCPv6 serveru.

- Router Advertisement zpráva v sekci options obsahuje Prefix Informations ve kterých je uvedena délka prefixu, která udává počet hlavních bitů. Počet bitů udává kolik hlavních bitů (zleva) bude rezervováno pro prefix, zbytek bude vyplněn nulami. Dále se zde nachází L a A flag. Hodnota bitu v L flagu udává, jestli prefix se vztahuje na linku (1), nebo ne (0). Pokud je hodnota A flagu 1 bit, značí to, že prefix může být využit pro SLAAC.



Obrázek 2.6: Sekce Prefix Informations obsažená v RA zprávě, převzato z [19]

- Host si vytvoří globální unicastovou IPv6 adresu spojením prefixu a identifikátoru rozhraní, který byl potvrzen jako unikátní u předešlé link-local adresy. Před přiřazením adresy se však ještě ověřuje její jedinečnost v síti pomocí DAD. DAD probíhá stejným způsobem jako u ověření link-local adresy, tedy vyšle se Neighbor Solicitation zpráva, která má jako cílovou adresu nastavenou právě vytvořenou adresu. Host čeká zda se nějaký node ozve pomocí Neighbor Advertisementu, kde říká, že už takovou adresu vlastní. Je to velice nepravděpodobné, ale kdyby takový případ nastal je zapotřebí, aby operační systém vygeneroval nové IID a proces tak zopakoval.

7. Pro urychlení si host může vygenerovat vlastní link-local adresu a její jedinečnost ověřit paralelně s Router Advertisementem. Díky možnému zpoždění RA je to rychlejší, než kdyby procesy probíhaly individuálně [20] [21].

2.11 DUID

DUID je zkratka pro DHCP Unique Identifier. Každý DHCP klient či server má DUID. Klienti pomocí DUID dokážou identifikovat, který DHCP server jim poslal zprávu. DHCP stavové (stateless) servery používají DUID k rozeznávání klientů a pro výběr konfiguračních parametrů ve spojení s IA (Identity Association).

Klienti i servery musí používat DUID pouze k rozeznávání. DUID by se nemělo používat k analyzování klientovy link-layer adresy, protože je to nespolehlivé. Klient může mít jinou link-layer adresu, než měl v době, kdy si generoval DUID.

DUID se přenáší v sekci options neboli možnostech, protože může nabývat různé délky a zároveň není potřebné ve všech DHCP zprávách. DUID je navrženo tak, aby bylo jedinečné mezi všemi DHCP nody (uzly) a aby bylo stále pro konkrétní node. DUID by se nemělo nikdy měnit v průběhu času, pokud je to možné. Tím se myslí, že by se DUID nemělo měnit v závislosti na výměně hardwaru, nebo na změně virtuálního rozhraní. DUID musí být jednoduché pro vytvoření pro jakékoliv zařízení. Zařízení se mohou lišit, například některá zařízení nemají trvalé úložiště. V takovém případě udržení DUID není možné.

DUID se skládá ze 2 oktetů (oktet = bajt) určující typy kódu, reprezentovaného v síťovém bajtovém pořadí (MSB neboli Most Significant Bit se nachází vlevo). Následuje proměnlivý počet oktetů, které tvoří aktuální identifikátor. Délka DUID bez druhu kódu je od 1 do 128 oktetů.

Type	Description
1	Link-layer address plus time
2	Vendor-assigned unique ID based on Enterprise Number
3	Link-layer address
4	Universally Unique Identifier (UUID) [RFC6355]

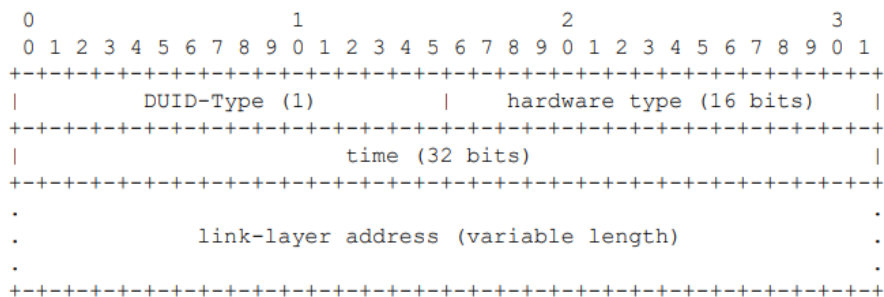
Obrázek 2.7: Typy DUID, převzato z [8]

Typy DUID:

1. **DUID vytvořeno na základě link-layer adresy a času (DUID-LLT)** - DUID-LLT se skládá ze 2 oktetů reprezentujících typ, v tomto případě se jedná o 1. typ (dle tabulky na obrázku č. 2.7). Následují 2 oktety, které obsahují kód typu hardwaru (u ethernetu se jedná například o link-layer adresu v kanonickém zápisu), 4 oktety obsahující časovou hodnotu (time value) ve formátu sekundy od půlnoci podle UTC (Coordinated Universal Time, neboli

Koordinovaný Světový Čas), měsíc, den, rok, modulo 2^{32} . Dále následuje link-layer adresa jakéhokoliv rozhraní, které je připojeno k DHCP zařízení.

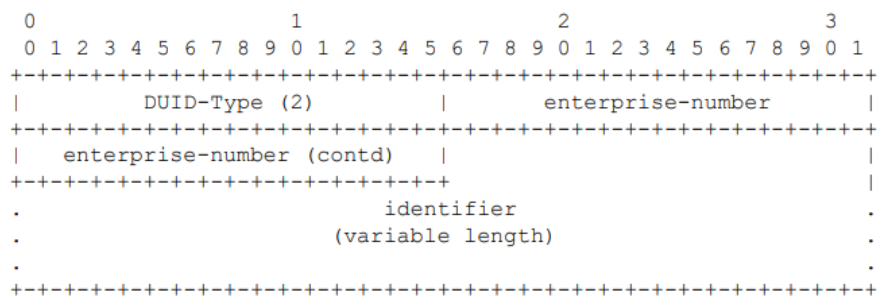
Tento typ DUID mohou využívat pouze zařízení, která mají trvalé úložiště (energeticky nezávislá paměťová jednotka). Tento typ DUID je doporučovaný pro běžná zařízení, jako jsou například PC, laptopy, tiskárny a routery. Díky časovému údaji je velice nepravděpodobná kolize dvou DUID. Kdyby ke kolizi však došlo, tak je zapotřebí, aby klient poskytl administrativní rozhraní, pomocí kterého nahradí kolizní DUID nově vytvořeným.



Obrázek 2.8: Formát DUID-LLT, převzato z [8]

2. **DUID přidělené prodejcem na základě čísla podniku (DUID-EN)** - Tento typ DUID je přidělen prodejcem. Skládá se ze 2 oktětů představující 2. typ DUID, ze 4 oktětů prodejcovy registračního čísla, které je udržováno IANA (Internet Assigned Numbers Authority, což je organizace, která celosvětově dohlíží na přidělování IP adres atd.). Následuje jedinečný identifikátor přidělený prodejcem.

DUID musí být přiřazeno při prvním použití zařízení a zároveň musí být uloženo na trvalém úložišti daného zařízení jako bezznaménkové (unsigned) 32 bitové číslo.



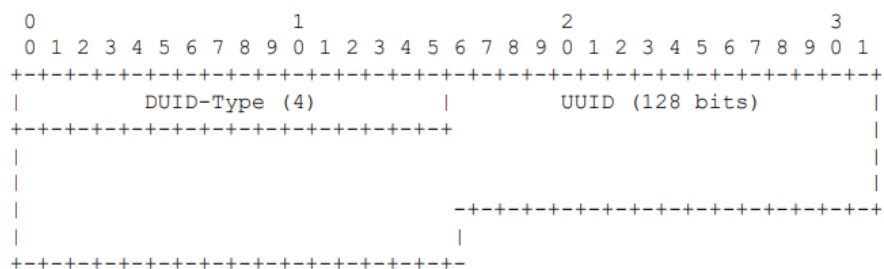
Obrázek 2.9: Formát DUID-EN, převzato z [8]

3. **DUID vytvořené na základě link-layer adresy (DUID-LL)** - Tento typ DUID tvoří 2 oktety 3. typu DUID, 2 oktety typu síťového hardwarového kódu (přidělen IANA), dále link-layer adresa jakéhokoliv rozhraní, které je trvale připojeno ke klientovi, nebo serveru. Může se jednat například o zabudovaný čip, který bude velmi nepravděpodobně odebrán a použit v

jiném zařízení. Právě taková zařízení mohou používat tento typ DUID. Formát DUID-LL je téměř identický s formátem DUID-LLT, pouze byla vynechána hodnota času.

4. **DUID vytvořeno na základě univerzálního jedinečného identifikátoru (DUID-UUID)** - Skládá se z 16 oktetů obsahující 128 bitové číslo UUID. UUID se využívá díky chybám předešlých DUID typů, které jsou podrobně popsány v RFC 6355. UUID verze 1 je vytvořeno na základě MAC adresy. Verze 3 až 5 jsou tvořeny na základě jména a pomocí něj vygenerovaných údajů, které pokaždé budou vygenerovány stejně.

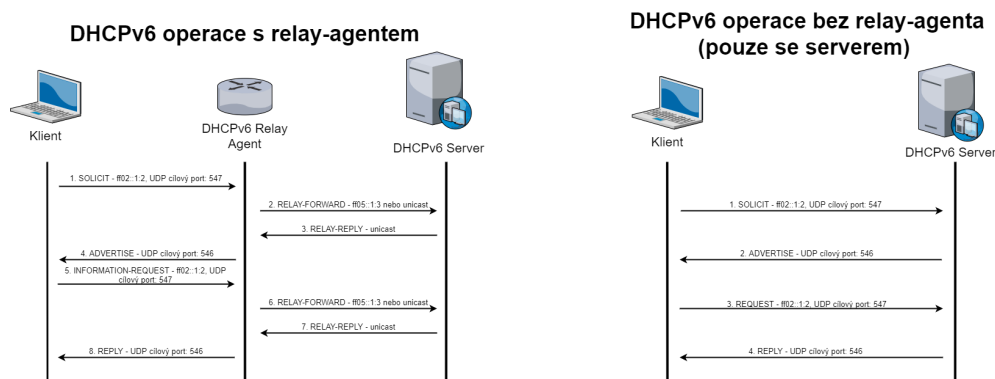
Algoritmus tvorby UUID probíhá získáním globálního zámku celého systému. Ze sdíleného úložiště celého systému přečte časové razítko (timestamp), sekvenci hodin (clock sequence) a ID nodu. Získá aktuální čas jako 60 bitový součet 100 nanosekundových intervalů od 00:00:00,00 15. října 1582, což je den přechodu ze čtvrtku 4. října juliánského kalendáře na pátek 15. října gregoriánského kalendáře. Získá se aktuální ID zařízení, pokud nebylo k dispozici, tak vygeneruje náhodnou časovou sekvenci. Pokud bylo ID k dispozici, ale uložené časové razítko je starší než aktuální zvýší, hodnotu časové sekvence. Algoritmus uloží stav aktuálního časového razítka, časové sekvence a ID zařízení. V RFC 4122 [22] je algoritmus podrobně popsán.



Obrázek 2.10: Formát DUID-UUID, převzato z [8]

2.12 Stateless DHCPv6

Celým názvem Dynamic Host Configuration Protocol version 6. Tato metoda se nazývá bezstavová (stateless), protože server neudrží žádné stavové informace klientů. DHCPv6 server v tomto případě pracuje pouze jako poskytovatel ostatních (other) konfiguračních parametrů, jako je seznam adres DNS rekurzivních serverů, SIP serverů, NTP serverů a informace pro síťové zavádění operačního systému. Neslouží k přidělování IPv6 adres. Aby host mohl používat stateless DHCPv6, musí mít adresu přidělenou jiným způsobem. K přidělení IPv6 adresy se nejčastěji používá již zmiňovaný SLAAC, nebo manuální konfigurace. Proto se ve většině případů využívá kombinace SLAACu a stateless DHCPv6. Pro bezstavové DHCPv6 je O flag nastaven na hodnotu 1 (O=1) a M flag je nastaven na 0 (M=0). Proces probíhá stejně jako u SLAACu počínaje NS. Změna nastává při Router Advertisementu, kde hodnota O flagu je 1. Poté začnou DHCPv6 operace, kde klient komunikuje se stateless DHCPv6 serverem a žádá ho o dodatečné informace.



Obrázek 2.11: Průběh DHCPv6 operací v závislosti na relay-agentovi

Průběh Stateless DHCPv6 operací [23]:

1. Po tom, co klientovi byla přidělena adresa pomocí SLAACu na základě Router Advertisementu, ve kterém měl O flag hodnotu 1, O flag dává klientovi na vědomí, že se v okolí nachází server, který mu může přidělit dodatečné informace. Klient nejdříve musí zjistit, kde se nachází DHCPv6 servery, proto pošle Solicit Message. Žádost je odeslána na cílový UDP port 547 a na adresu multicastové skupiny rezervované pro všechny DHCPv6 servery (FF02::1:2).
2. Všechny DHCPv6 servery, které souhlasí s požadavky klienta, pošlou klientovi Advertise message, který znamená, že daný server je dostupný pro DHCPv6 služby.
3. Klient si vybere jeden ze serverů. Protože se jedná o stateless DHCPv6 server, klient pošle DHCPv6 Information-Request Message DHCPv6 serveru (u statefull DHCPv6 se používá Request Message). Požaduje pouze konkrétní konfigurační parametry, jako jsou adresy rekurzivních DNS serverů a SIP serverů.
4. DHCPv6 server pošle klientovi DHCPv6 Reply Message obsahující informace, o které si klient zažádal.
5. Pokud by se jednalo o případ, kde mezi klientem a DHCPv6 serverem je relay-agent, tak se počet zpráv zdvojnásobí, protože relay-agent zde slouží pouze jako prostředník a veškerou komunikaci přeposílá dál. Zpráva Relay-Forward se využívá ve směru od klienta k serveru a Relay-Reply v opačném, tedy od serveru ke klientovi.

2.13 Statefull DHCPv6

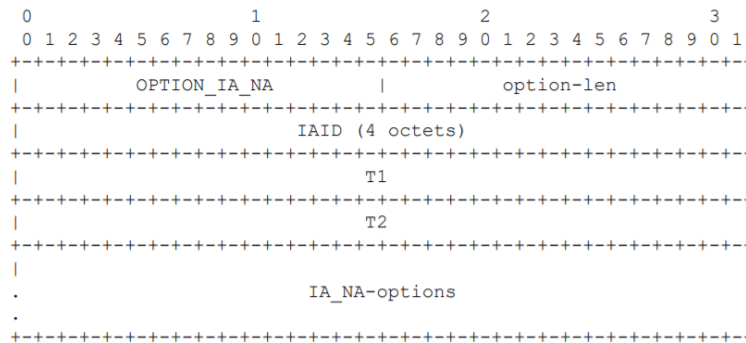
Tato metoda je nejpodobnější běžnému DHCPv4. U této metody RA obsahuje prefix, délku prefixu a M-bit ($M = 1$). M-bit klientovi říká, aby nepoužíval prefix, ale že mu bude IPv6 adresa přidělena

pomocí stavového DHCPv6 serveru. Tato metoda se nazývá statefull, protože server udržuje IPv6 stavové informace o klientovi v databázi leases.

Identity Association - K označení se používá zkratka IA. IA je konstrukce, pomocí které server a klient identifikují a organizují sady souvisejících IPv6 adres nebo delegovaných prefixů. IA se skládá z IAID a přiřazených konfiguračních informací. IAID je unikátní pro každý typ IA (IAID 0 pro IA_NA je unikátní, přestože IA obsahuje další záznam s IAID 0 pro IA_PD).

Rozdělení IA na základě poskytnutých informací:

- **Identity Association pro přidělování adres** - Každé rozhraní, které si zažádá o přidělení adresy pomocí stavového DHCPv6 serveru, musí mít klientem přiřazene IA. Klient tak přiřazené IA využívá k získání konfiguračních informací. Každé takové IA musí být spojováno právě s jedním rozhraním.
 - IA_TA - IA pro dočasné adresy (temporary addresses) - Daly by se označit, jako jednorázové adresy, protože je doporučeno, aby se neobnovovaly ani neprodlužovaly (renew, rebind). Jejich životnost (lifetime) je zapouzdřena v sekci options a nedoporučuje se, aby intervaly byly delší než TEMP_VALID_LIFETIME a TEMP_PREFERRED_LIFETIME, které mají defaultní hodnotu 1 týden a 1 den [12].
 - IA_NA - IA pro adresy, které nejsou dočasné (non-temporary addresses) - Má proti IA_TA navíc T1 a T2 časové intervaly, po kterých klient musí požádat o prodloužení životnosti (lifetime) zapůjčených adres. Je to čas v sekundách kdy životnost expiruje. Jakmile T1 vyprší, tak klient pošle žádost o prodloužení (renew). Pokud klient nedostane žádnou odpověď, tak po vypršení delšího časového intervalu T2 pošle opět zprávu, kterou žádá o prodloužení životnosti adres (rebind). Klient renew nebo rebind zprávy neposílá ihned, ale pouze ve vymezených časech (renew, rebind transmission times), aby se zamezilo paketovým bouřím.

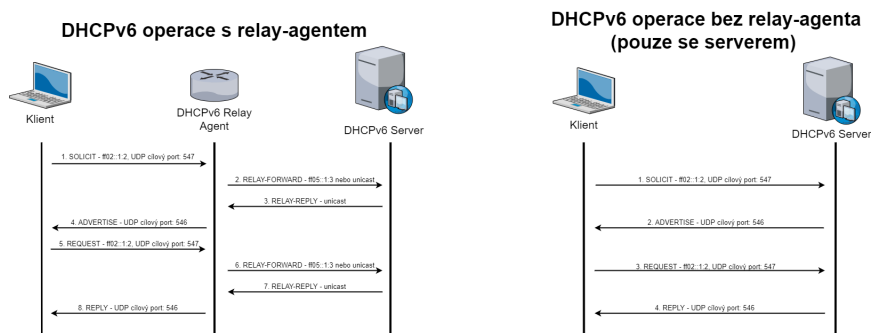


Obrázek 2.12: Formát IA_NA, převzato z [8]

- **Identity Association pro delegaci prefixů** - IA_PD se od IA pro přidělování adres liší tím, že neposkytuje adresy, ale poskytuje prefix, tzv. address pool. Používá se při rozsítování ISP (Internet Service Provider), kdy relay-agent deleguje prefixy odběratelům pomocí jinému routeru, který se stará o určitou podsíť. ISP tak dělá z /48 prefixů /64 pro každou linku. IA nemusí být spojováno právě s jedním rozhraním, ale může být přiřazeno ke klientovi se sadou rozhraní. Prefixy poskytnuté DHCPv6 serverem klientovi také obsahují T1 a T2 časové intervaly.

Průběh Statefull DHCPv6 operací:

- RS i RA probíhá stejně, jako u stateless DHCPv6 operací, kdy klient při tvorbě RS vygeneruje transaction-id, které dále používají po celou dobu komunikace se serverem. Důležité jsou i DUID, která se používají, aby servery s jiným DUID zprávu dále nezpracovávaly, ale ihned zahodily. Změna nastává až u DHCPv6 žádosti, kde se u bezstavového DHCPv6 serveru používá Information Request.
- Pokud v RA nabývá M-bit hodnotu 1, klient pošle Request Message DHCPv6 serveru, požadující IPv6 adresu a další konfigurační parametry jako je DNS a jméno domény, nebo žádá o delegaci prefixů.
- DHCPv6 server pošle klientovi Reply Message obsahující informace, o které si klient zažádal.
- Celkově existuje 13 typů DHCPv6 zpráv, které jsou popsány v RFC 8415 v sekci 7.3. Ve stejném RFC je v sekci 18.2. podrobně popsáno chování klienta, požadavky, které musí být splněny pro jednotlivé zprávy a kde jsou uvedeny jednotlivé informace v jednotlivých DHCPv6 zprávách [8].



Obrázek 2.13: Průběh stavových DHCPv6 operací v závislosti na relay-agentovi

Kapitola 3

Popis grafického síťového simulátoru GNS3

GNS3 je grafický síťový simulátor, který byl poprvé vydán v roce 2008. Stejně jako více rozšířený a známý Cisco Packet Tracer slouží k simulaci počítačových sítí. Rozdíl mezi CPT a GNS3 je, že GNS3 využívá virtuálních zařízení, která běží na vašem počítači pomocí jiných programů určených k virtualizaci. GNS3 je volně dostupný (open source free software). GNS3 poskytuje velké množství síťových emulačních prostředků jako jsou routery atd.

Toto prostředí je vhodné pro testování konfigurací topologií, které mají být uvedeny do provozu. Zamezí se tak výpadkům, nebo jiným nepříjemným problémům. Další možností využití GNS3 prostředí může být z edukativního hlediska. Tuto možnost mohou využít například studenti připravující se na různé certifikační zkoušky ze síťových a komunikačních technologií, jako jsou například Cisco CCNA, CCNP apod. [24].

3.1 Instalace a příprava GNS3 prostředí

GNS3 lze stáhnout přímo z oficiálních webových stránek [25]. Instalace freewareového prostředí GNS3 proběhlo bez problémů. Při seznamování s prostředím jsem zjistil, že zde není možnost pro simulaci routerů, proto jsem si dohledal, že IOSy jednotlivých routerů si musím nainstalovat manuálně. Při pokusu o instalaci routerů jsem zjistil, že ne všechny podporují Windows a proto je potřeba nainstalovat Virtual Machine, buďto v Oracle VirtualBoxu, nebo VMWare Workstation, který GNS3 doporučuje, protože údajně nezabírá tolik RAM. Proto jsem tedy nainstaloval verzi VMWare Player, která je zdarma pro nekomerční účely, ale bohužel jsem se setkal s problémem díky nedostatku místa na disku C. Byl jsem nucen nainstalovat VMWare na disk D. Dále podle návodu jsem si stáhnul VM pro VMWare z oficiálních stránek. Po naimportování jsem však v GNS3 v "Edit/preferences" v části GNS3 VM zvolil virtualizační prostředí "Virtualization Engine", ale nedokázal jsem vybrat konkrétní zařízení "VM name". Hledal jsem řešení tohoto problému na různých fórech, vyzkoušel jsem několik řešení, jako například upřesnit cestu k VM v konfiguračním souboru umístěném v C:\Users\jakub\AppData\Roaming\VMware\preferences.ini, ale žádné z řešení nepomohlo. Proto

jsem vyzkoušel druhé virtualizační prostředí Oracle VirtualBox, které jsem taktéž umístil na disk D a nesetkal jsem se s předešlým problémem. Jako jiné emulátory lze využít Dynamips sloužící pro Cisco routery, Qemu sloužící pro firewally a Linuxové stanice.

Dále jsem si na oficiálním webu GNS3 v Appliances [26] našel vhodné zařízení k realizaci mé topologie. Je důležité si uvědomit, že některé z nich při importování budou požadovat licence, které lze zakoupit za velmi vysokou částku. Mezi IOSy vyžadující licenci se řadila většina zařízení Cisco.

3.2 Grafické prostředí

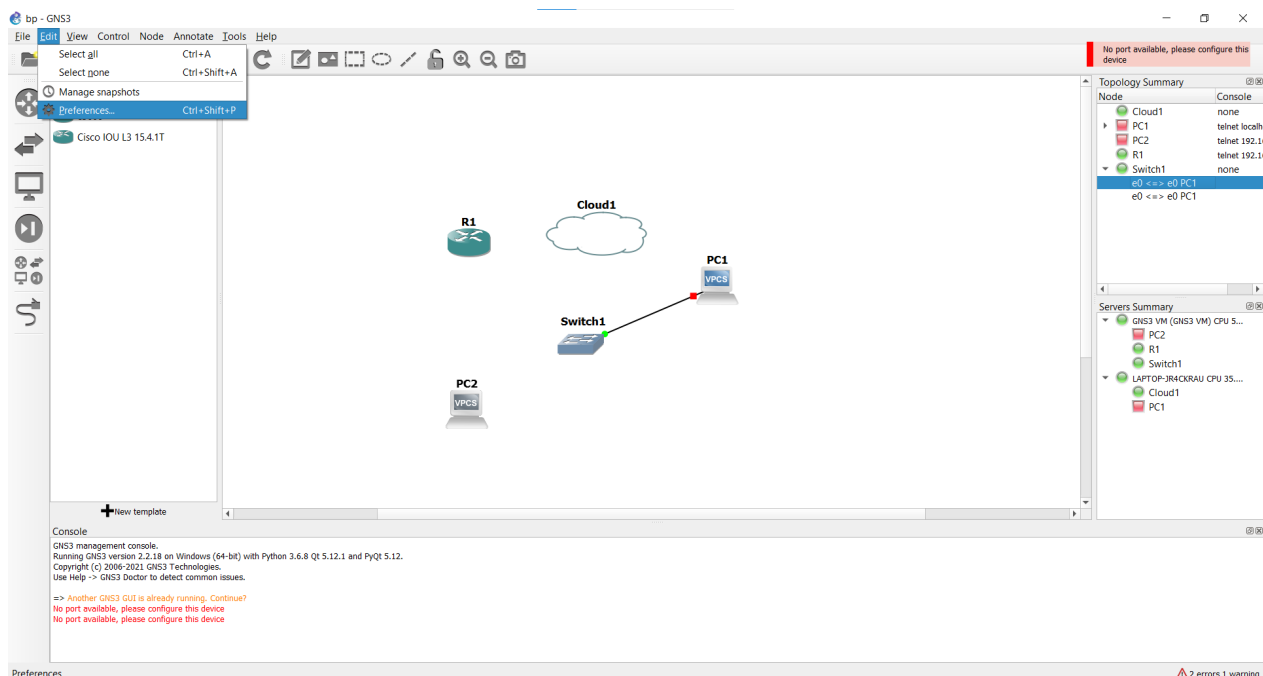
GNS3 má velmi přívětivé a přehledné grafické prostředí i pro nového uživatele, viz obrázek č. 3.1. Na levé straně se nachází paleta s výběrem zařízení seřazených podle funkce a vrstvy, na které pracují. Paleta shora dolů obsahuje:

- Routery
- Switche
- Koncová zařízení
- Bezpečnostní zařízení
- Všechna zařízení
- Spojení (link)

Na pravé straně lze vidět shrnutí topologie, kde lze pomocí barevných signálů zjistit stav jednotlivých nodů (uzlů). Červený čtvereček znamená, že zařízení není aktivní a zelené kolečko znamená, že zařízení běží. Po rozkliknutí či rozbalení jednotlivých zařízení lze vidět jejich rozhraní a kam daná linka vede. Níže je umístěno shrnutí serverů, kde taktéž lze vidět stav serveru a po rozkliknutí lze vidět, které zařízení běží na vašem lokálním počítači, nebo na VM.

V horní části v možnosti "Edit" se nachází položka "preferences", kde lze najít většinu nastavení a voleb ohledně VM. Další užitečnou položkou možností je zobrazení názvů rozhraní "Show/Hide interface labels", kterou lze nalézt ve "View".

V dolní části je konzole, která vypisuje případné chyby.



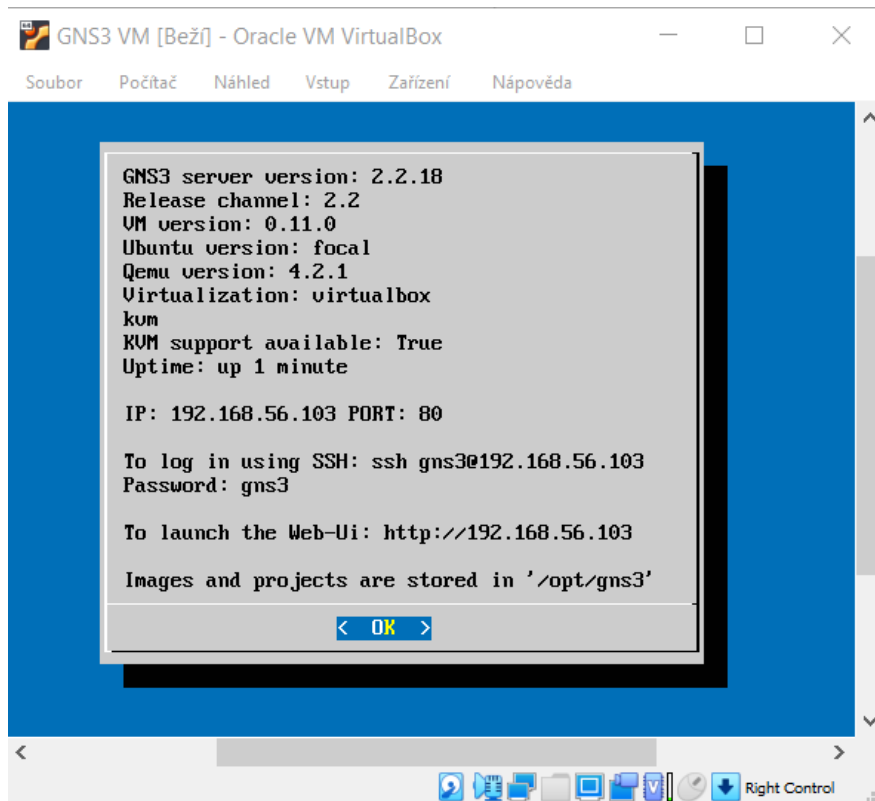
Obrázek 3.1: GNS3 grafické prostředí

3.3 Virtual Machine

K instalaci VM je zapotřebí virtuální prostředí, já používal Oracle VM VirtualBox verze 6.1.14.

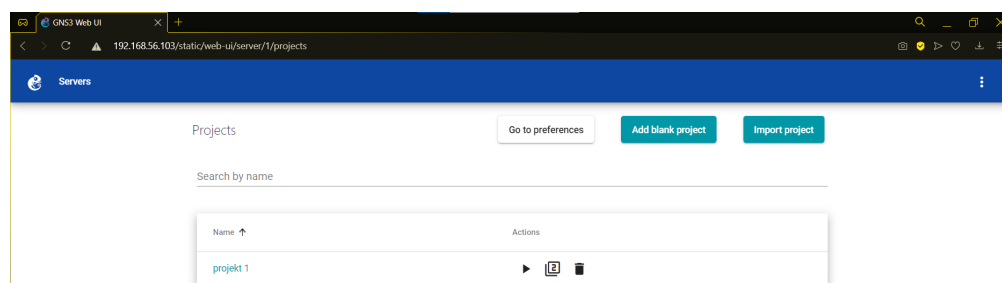
Jak jsem již výše uvedl, je potřeba nainstalovat GNS3 VM, aby bylo dále možné instalovat některá zařízení, která nejsou kompatibilní s operačním systémem Windows, což by mohlo vést k různým chybám.

Na oficiálních stránkách [27] jsem si stáhnul VM pro VirtualBox. Je velice důležité, aby se verze shodovaly. Já jsem použil verzi 2.2.18. Při práci nedoporučuji aktualizovat GNS3, protože poté budete také muset nainstalovat nové VM pro novou verzi, což by vás mohlo podstatně zdržet. Pro spuštění serveru v GNS3 musí být zvolen virtualizační modul "Virtualization Engine" a jeho název "VM name", toto nastavení se nachází v Preferences, GNS3 VM.



Obrázek 3.2: GNS3 VM ve VirtualBoxu

Na obrázku č. 3.3 lze vidět IP adresu 192.168.56.103, kterou lze zadat do webového prohlížeče. Tím se zpřístupní webového prostředí GNS3, kde lze vytvářet projekty a pracovat se základními zařízeními. Je zde i možnost importovat VirtualBox, nebo jiné templaty, ale to jsem osobně nezkoušel.

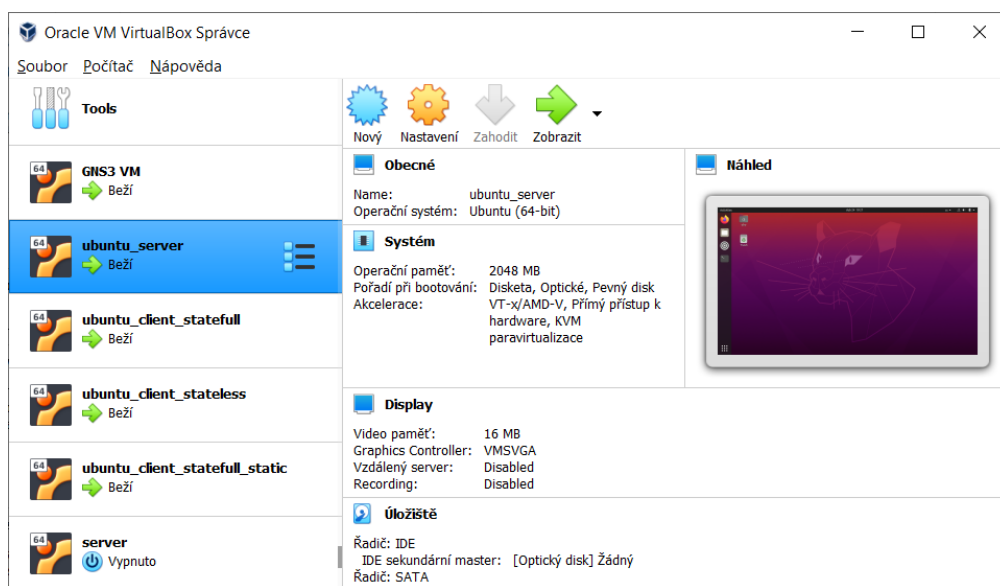


Obrázek 3.3: GNS3 webové prostředí v prohlížeči

3.4 Instalace Ubuntu ve VirtualBoxu

Pro realizaci DHCPv6 serveru i pro klientské zařízení jsem se rozhodl využít Ubuntu Desktop verze 20.04.2.0 LTS, který jsem si stáhnul z oficiálních stránek [28] a poté ve VirtualBoxu vybral "Nový",

dále jsem specifikoval název podle účelu, jako typ jsem zvolil Linux a verzi Ubuntu (64-bit). V dalším kroku jsem zvolil velikost paměti RAM 1024 MB, ale později při práci mi VM přišly pomalé, proto jsem zdvojnásobil velikost paměti RAM na 2048 MB, což bych všem doporučil, pokud mají dostatečně výkonný počítač. Pokračoval jsem s výchozími volbami "Vytvořit nyní virtuální pevný disk", "VDI (VirtualBox Disk Image)", "Dynamicky alokované", 32 GB. Po vytvoření jsem označil daný VM a otevřel si jeho nastavení, kde jsem v sekci úložiště vybral Řadič: IDE, kde jsem zvolil již stažený Ubuntu instalační .iso soubor. Zkontroloval jsem zda budu mít připojení k internetu v sekci "Síť", kde jsem přidal další síťovou kartu připojenou k síťovému mostu a v pokročilém nastavení jsem zvolil "Promiskuitní režim: Povoleno vše". Před dokumentací jsem doplnil sdílené úložiště pro přesun textových souborů obsahující konfigurace, ale to pro samotnou konfiguraci není potřebné. Nastavení jsem uložil a spustil zařízení. Je zapotřebí projít instalačním procesem, kde je potřeba definovat věci jako jazyk, časovou zónu a podobně. Déle jsem si přizpůsobil Ubuntu tak, aby se mi s ním dobře pracovalo, například jsem zvolil vhodné rozlišení, přidal si anglickou klávesnici atd.



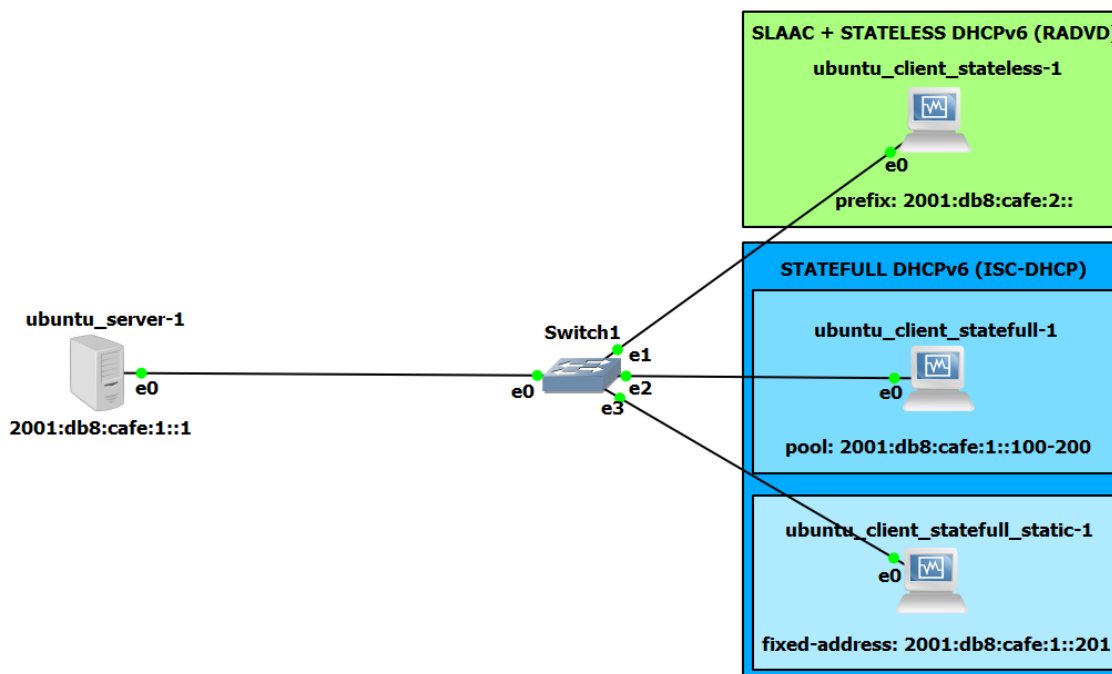
Obrázek 3.4: Nainstované Ubuntu VM ve VirtualBoxu

Aby nainstalované Ubuntu VM šlo používat v GNS3 je zapotřebí ho zde přidat, což lze opět udělat v preferencích pod názvem VirtualBox VMs pomocí tlačítka "New", kde v rozbalovacím seznamu vyberete název VM a potvrdíte. Poté, aby šlo Ubuntu spojit s jiným zařízením a nevysklovala chybová hláška, musí být upraveno nastavení pomocí tlačítka "Edit", "Network", kde musí být zaškrtnuto políčko "Allow GNS3 to use any configured VirtualBox adapter".

Kapitola 4

Návrh zapojení pro testování automatického přidělování IPv6 adres

Ke konfiguraci jsem si vybral velmi jednoduchou topologii, která se skládá z jednoho switchu a 4 Ubuntu VM, kde jedno slouží jako DHCPv6 server a zbylé 3 jako klienti, viz obrázek č. 4.1. Klienti budou mít přiřazenou adresu, každý jiným způsobem. Klient *ubuntu_client_stateless-1* bude mít přidělenou adresu pomocí SLAACu díky Router Advertisementům, které budou rozesílány díky RADVD na serveru. Zbývající klienti budou mít přiděleny adresy pomocí statefull DHCPv6, které je zprostředkované na serveru pomocí ISC DHCP. Prostřední klient *ubuntu_client_statefull-1* dostane adresu z DHCPv6 poolu IPv6 adres 2001:db8:cafe:1::100 až 2001:db8:cafe:1::200. Klientovi *ubuntu_client_statefull_static-1* nacházejícímu se v topologii nejnižší bude přidělena fixní adresa, kterou server na základě klientova ID určil jako 2001:db8:cafe:1::201.



Obrázek 4.1: Topologie v prostředí GNS3

4.1 Konfigurace Ubuntu sloužící jako server

Je důležité si uvědomit, že jakmile spojíme jednotlivá zařízení, tak nebudou mít přístup k internetu. Proto jsem si pomocí následujících příkazů v terminálu stáhnul net-tools, abych mohl používat *ifconfig*. Dále jsem si stáhnul ISC DHCP server daemon, který bude sloužit jako stavové DHCPv6, a RADVD daemon pro odesílání zpráv Router Advertisement.

- `sudo nano apt-get update`
- `sudo nano apt install net-tools`
- `sudo nano apt install isc-dhcp-server`
- `sudo nano apt install radvd`

Pomocí příkazu *ifconfig* jsem si zjistil, že budu konfigurovat rozhraní *enp0s3*. Rozhraní lze také zjistit příkazem *ip address*. Dalším příkazem jsem vypnul Network-Manager, který nedoporučuji používat na zařízení, které má sloužit jako server.

- `sudo service network-manager stop`

Provedl jsem manuální konfiguraci IPv6 adresy.

- `sudo ip -6 address add 2001:cafe:cafe:cafe::1/64 dev enp0s3`

Konfigurace ISC DHCP:

Zálohoval a upravil jsem konfigurační soubor v textovém editoru *nano* určeného pro terminál.

- `sudo cp /etc/dhcp/dhcpd6.conf /etc/dhcp/dhcpd6__zaloha.conf`
- `sudo nano /etc/dhcp/dhcpd6.conf`

V konfiguračním souboru, který je velmi dobře a přehledně předchystaný, jsem využil zakomentovaných, nebo předepsaných částí a ty jsem následovně změnil:

```
option dhcp6.name-servers 2001:4860:4860::8888, 2001:4860:4860::8844;  
option dhcp6.domain-search "dns.dzi0027.cz","dzi00227.cz";
```

Čímž jsem nastavil globální definici možných DNS serverů a doménových jmen.

```
host u_sful_static {  
    host-identifier option dhcp6.client-id 00:01:00:01:28:15:a0:ca:08:00:27:f0:b1:7c;  
    hardware ethernet 08:00:27:f0:b1:7c;  
    fixed-address6 2001:db8:cafe:1::201;  
}
```

Tímto jsem nastavil, že klient se zadaným identifikátorem bude mít přidělenou danou fixní adresu. Jako `client-id` jsem nastavil defaultní DUID 3. klienta.

```
subnet6 2001:db8:cafe:1::/64 {  
    range6 2001:db8:cafe:1::100 2001:db8:cafe:1::200;  
    option dhcp6.name-servers 2001:4860:4860::8888, 2001:4860:4860::8844;  
    option dhcp6.domain-search "dns.dzi0027.cz","dzi0027.cz";  
}
```

V této konfiguraci `subnet6` definuje podsít, `range6` definuje rozsah adres pro klienty. Zbytek konfigurace stejně jako v globální části definují DNS servery a doménová jména pro tento DHCPv6 pool.

Po nakonfigurování jsem službu zapnul a zkontroloval stav serveru.

- `sudo service isc-dhcp-server6 start`
- `sudo service isc-dhcp-server6 status`

Spuštění serveru se nepodařilo, což bylo ohlášeno chybovou hláškou "failed". Proto jsem si nechal vypsat podrobnější chybové hlášení.

- `sudo cat /var/lib/syslog`

Zde bylo napsáno, že nejsou definována rozhraní. Proto jsem je nastavil následujícím způsobem. Zaprvé jsem otevřel konfigurační soubor.

- `sudo nano /etc/default/isc-dhcp-server`

V tomto souboru jsem odkomentoval následující řádky a doplnil rozhraní pro IPv6.

```
DHCPDv6_CONF=/etc/dhcp/dhcpd6.conf
DHCPDv6_PID=/etc/dhcp/dhcpd6.pid
INTERFACESv6="enp0s3"
```

Konfigurace RADVD:

RADVD po instalaci automaticky nevytvořilo konfigurační soubor jako ISC DHCP, proto jsem si musel vytvořit nový [29].

- `sudo nano /etc/radvd.conf`

Do tohoto konfiguračního souboru jsem vložil následující konfiguraci, která pro rozhraní `enp0s3` povoluje odesílání Router Advertisementů s Other i Managed flagem ($M = 1$, $O = 1$). Definuje minimální a maximální časový interval Router Advertisementů a prefix. Dále konfigurace upřesňuje jak má být prefix používán. Povoluje využití prefixu pouze na této lince. Povoluje prefix využít ke SLAACu, díky kterému klient získá adresu. Jedná se o L a A flag, které jsou podrobněji popsány v teoretické části. Poslední možnost *AdvRouterAddr* je vypnutá, protože by při zapnutí znamenala, že bude odesílána adresa rozhraní místo prefixu, což v tomto případě nechceme.

```
interface enp0s3
{
  AdvSendAdvert on;
  AdvManagedFlag on;
  AdvOtherConfigFlag on;
  MinRtrAdvInterval 30;
  MaxRtrAdvInterval 100;
  prefix 2001:db8:cafe:2::/64
  {
    AdvOnLink on;
    AdvAutonomous on;
    AdvRouterAddr off;
  };
}
```

```
};
```

RADVD jsem po nakonfigurování taktéž zapnul a zkontroloval jeho stav.

- `sudo service radvd start`
- `sudo service radvd status`

V tomto případě spuštění proběhlo úspěšně a RADVD daemon byl aktivní, ale vypisoval hlášku "IPv6 forwarding setting is: 0, should be 1 or 2". Tuto chybu jsem opravil následujícím příkazem. RADVD jsem restartoval a po výpisu stavu se již chybová hláška nevyskytovala.

- `sudo sysctl -w net.ipv6.conf.all.forwarding=1`
- `sudo service radvd restart`
- `sudo service radvd status`

4.2 Nastavení stateless Ubuntu klienta

V terminálu jsem pomocí network manager command line tool vytvořil síťový profil, ve kterém jsem nastavil rozhraní a specifikoval metodu, kterou získá IPv6 adresu. Nastavení jsem si ověřil a nastavil síťový profil jako aktivní. Síťový profil lze zobrazit i v grafickém prostředí v nastavení v záložce "Network".

- `sudo nmcli connection add con-name autokonfigurace ifname enp0s3 type ethernet`
- `sudo nmcli connection modify autokonfigurace ipv6.method auto`
- `nmcli connection show autokonfigurace`
- `sudo nmcli connection up autokonfigurace`

Pokud by se stalo, že při kontrole nebyl přiřazen DNS server a doménové jméno, lze si o něj požádat manuálně pomocí *dhclient* s přepínačem `-S`, který udává přidělení pouze stateless konfiguračních parametrů na základě odeslané zprávy `information-request`. To znamená, že nebude vytvořen zápis v lease databázi.

- `sudo cat /run/systemd/resolve/resolv.conf`
- `sudo dhclient -6 -S enp0s3`

V konfiguračním souboru *dhclient* lze specifikovat, které konfigurační parametry si zažádá.

- `sudo nano /etc/dhcp/dhclient.conf`

V konfiguračním souboru jsem smazal v sekci request vše, kromě:

```
subnet-mask, domain-name, host-name, dhcp6.name-servers, dhcp6.domain-search;
```

4.3 Nastavení statefull Ubuntu klienta

Stejně jako u předešlého klienta jsem vytvořil síťový profil, ve kterém jsem nastavil DHCP metodu přidělování IPv6 adresy.

- `sudo nmcli connection add con-name moje-dhcp ifname enp0s3 type ethernet`
- `sudo nmcli connection modify moje-dhcp ipv6.method dhcp`
- `nmcli connection show moje-dhcp`
- `sudo nmcli connection up moje-dhcp`

Následovně jsem z konfiguračního souboru klienta vymazal věci, které se v mé topologii nevy-
skytují, např sntp server, ntp server atd.

4.4 Nastavení statefull fixního Ubuntu klienta

Tento klient bude mít přidělenou adresu DHCPv6 serverem stejně jako předešlý klient, proto na-
stavení síťového profilu bude úplně totožné.

Důležitá část přichází právě v konfiguračním souboru klienta, kde je zapotřebí definovat, aby při
žádosti odesílal klientův identifikátor, který jsem definoval jako jeho defaultní DUID, stejně jako
při konfiguraci serveru.

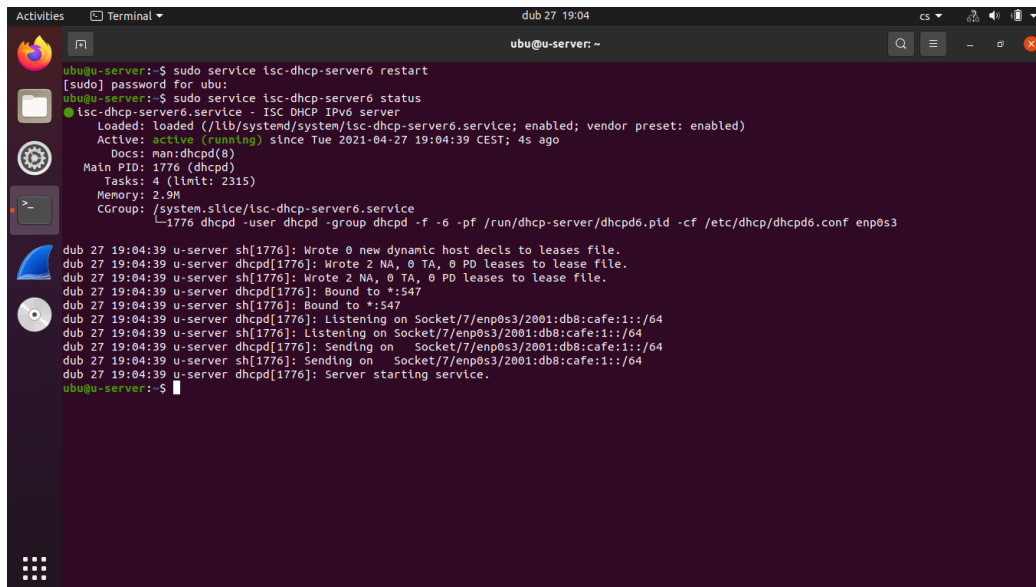
```
send dhcp-client-identifier 00:01:00:01:28:15:a0:ca:08:00:27:f0:b1:7c;
```

Server podle tohoto identifikátoru rozezná klienta a přiřadí mu definovanou fixní IPv6 adresu.

4.5 Ověření navrženého řešení

Ověření funkčnosti konfigurace serveru:

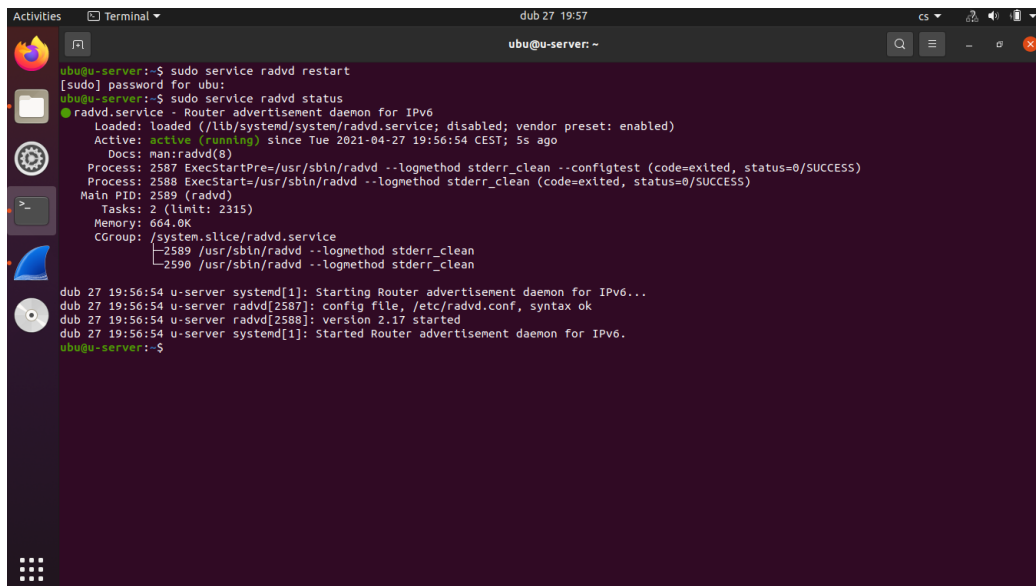
Na obrázcích č. 4.2 a 4.3 lze vidět ověření zprovoznění ISC DHCPv6 a RADVD daemon služeb. Důkazem je zelený nápis *active (running)*



```
ubu@u-server:~$ sudo service isc-dhcp-server6 restart
[sudo] password for ubu:
ubu@u-server:~$ sudo service isc-dhcp-server6 status
● isc-dhcp-server6.service - ISC DHCP IPv6 server
   Loaded: loaded (/lib/systemd/system/isc-dhcp-server6.service; enabled; vendor preset: enabled)
   Active: active (running) since Tue 2021-04-27 19:04:39 CEST; 45s ago
     Docs: man:dhcpd(8)
    Main PID: 1776 (dhcpd)
      Tasks: 4 (limit: 2315)
     Memory: 2.9M
    CGroup: /system.slice/isc-dhcp-server6.service
            └─1776 dhcpd -user dhcpd -group dhcpd -f -6 -pf /run/dhcp-server/dhcpd6.pid -cf /etc/dhcp/dhcpd6.conf enp0s3

dub 27 19:04:39 u-server sh[1776]: Wrote 0 new dynamic host decls to leases file.
dub 27 19:04:39 u-server dhcpd[1776]: Wrote 2 NA, 0 TA, 0 PD leases to lease file.
dub 27 19:04:39 u-server sh[1776]: Wrote 2 NA, 0 TA, 0 PD leases to lease file.
dub 27 19:04:39 u-server dhcpd[1776]: Bound to *:547
dub 27 19:04:39 u-server sh[1776]: Bound to *:547
dub 27 19:04:39 u-server dhcpd[1776]: Listening on Socket/7/enp0s3/2001:db8:cafe:1::/64
dub 27 19:04:39 u-server sh[1776]: Listening on Socket/7/enp0s3/2001:db8:cafe:1::/64
dub 27 19:04:39 u-server dhcpd[1776]: Sending on Socket/7/enp0s3/2001:db8:cafe:1::/64
dub 27 19:04:39 u-server sh[1776]: Sending on Socket/7/enp0s3/2001:db8:cafe:1::/64
dub 27 19:04:39 u-server dhcpd[1776]: Server starting service.
ubu@u-server:~$
```

Obrázek 4.2: Databáze leases serveru

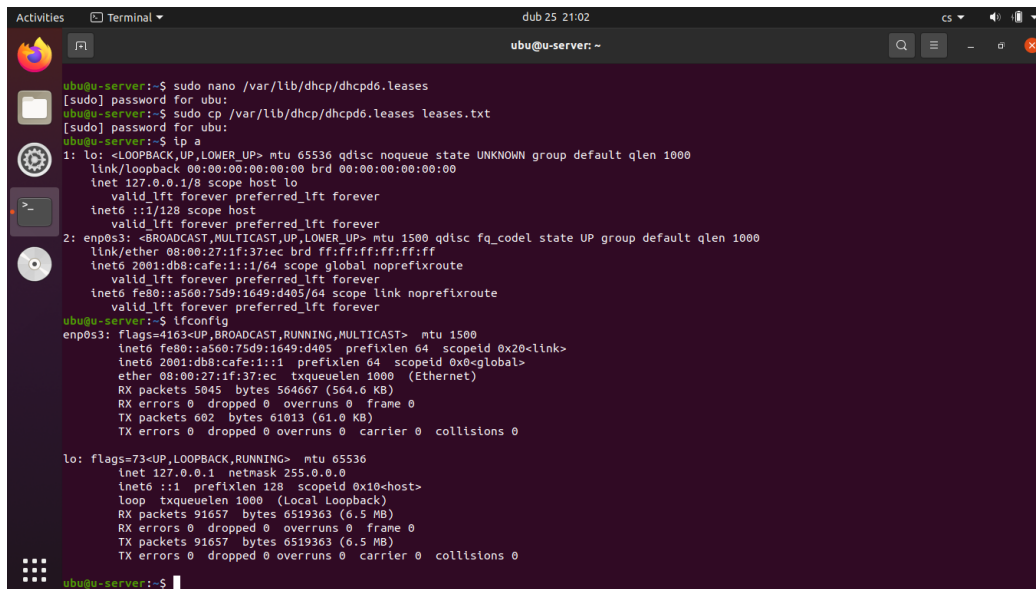


```
ubu@u-server:~$ sudo service radvd restart
[sudo] password for ubu:
ubu@u-server:~$ sudo service radvd status
● radvd.service - Router advertisement daemon for IPv6
   Loaded: loaded (/lib/systemd/system/radvd.service; disabled; vendor preset: enabled)
   Active: active (running) since Tue 2021-04-27 19:56:54 CEST; 5s ago
     Docs: man:radvd(8)
    Process: 2587 ExecStartPre=/usr/sbin/radvd --logmethod stderr_clean --configtest (code=exited, status=0/SUCCESS)
    Process: 2588 ExecStart=/usr/sbin/radvd --logmethod stderr_clean (code=exited, status=0/SUCCESS)
    Main PID: 2589 (radvd)
      Tasks: 2 (limit: 2315)
     Memory: 664.0K
    CGroup: /system.slice/radvd.service
            └─2589 /usr/sbin/radvd --logmethod stderr_clean
              └─2590 /usr/sbin/radvd --logmethod stderr_clean

dub 27 19:56:54 u-server systemd[1]: Starting Router advertisement daemon for IPv6...
dub 27 19:56:54 u-server radvd[2587]: config file, /etc/radvd.conf, syntax ok
dub 27 19:56:54 u-server radvd[2588]: version 2.17 started
dub 27 19:56:54 u-server systemd[1]: Started Router advertisement daemon for IPv6.
ubu@u-server:~$
```

Obrázek 4.3: Databáze leases serveru

Následující obrázek č. 4.4 obsahuje výpis příkazů *ip address* a *ifconfig*, kde lze vidět staticky nastavenou IPv6 adresu. Na obrázku č. 4.5 lze vidět databáze leases obsahující IA_NA záznamy, ze kterých lze vyčíst datum a čas přidělení adresy, přidělenou IPv6 adresu, stav, lifetime a čas vypršení platnosti. V databázi leases serveru nenaleznete žádné záznamy o fixních adresách, které jsou mimo adresní prostor, protože nemohou být přiděleny nikomu jinému, pouze klientovi s patřičným identifikátorem. Celý soubor lze najít v příloze 4.



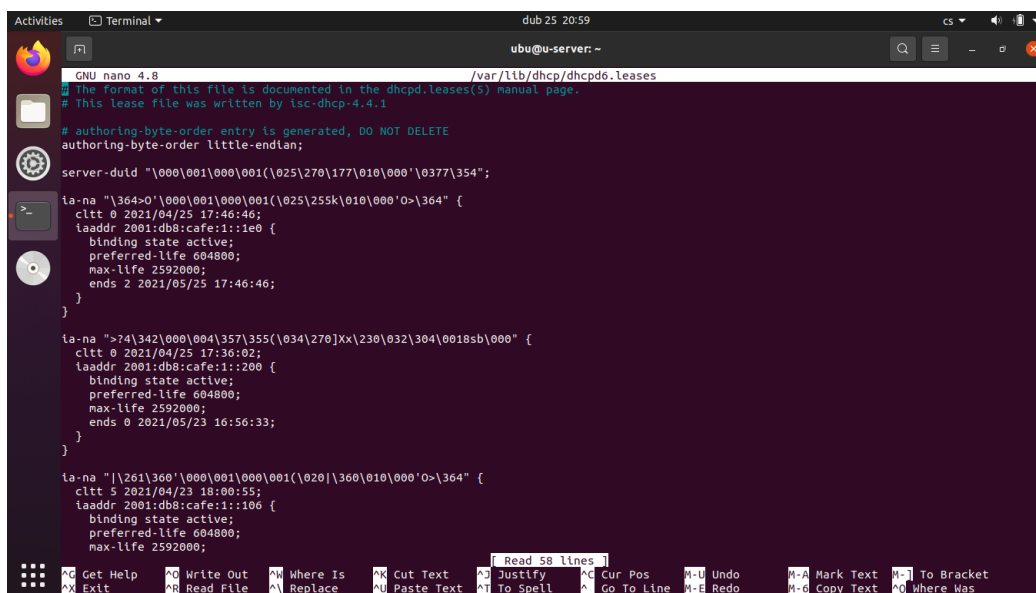
```

ubu@u-server:~$ sudo nano /var/lib/dhcp/dhcpd6.leases
[sudo] password for ubu:
ubu@u-server:~$ sudo cp /var/lib/dhcp/dhcpd6.leases leases.txt
[sudo] password for ubu:
ubu@u-server:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:1f:37:ec brd ff:ff:ff:ff:ff:ff
    inet6 2001:db8:cafe:1::1/64 scope global noprefixroute
        valid_lft forever preferred_lft forever
    inet6 fe80::a560:75d9:1649:d405/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
ubu@u-server:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet6 fe80::a560:75d9:1649:d405 prefixlen 64 scopeid 0x20<link>
    inet6 2001:db8:cafe:1::1 prefixlen 64 scopeid 0x0<global>
    ether 08:00:27:1f:37:ec txqueuelen 1000 (Ethernet)
    RX packets 5045 bytes 564667 (564.6 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 602 bytes 61013 (61.0 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 91657 bytes 6519363 (6.5 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 91657 bytes 6519363 (6.5 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

```

Obrázek 4.4: Výpis IP adresy serveru



```

GNU nano 4.8 /var/lib/dhcp/dhcpd6.leases
# The format of this file is documented in the dhcpd.leases(5) manual page.
# This lease file was written by ISC-dhcp-4.4.1

# authoring-byte-order entry is generated, DO NOT DELETE
authoring-byte-order little-endian;

server-duid "\000\001\000\001\025\270\177\010\000\0377\354";

ia-na "\364\0\000\001\000\001\025\255k\010\000\0\364" {
    cltt 0 2021/04/25 17:46:46;
    iaaddr 2001:db8:cafe:1::1e0 {
        binding state active;
        preferred-life 604800;
        max-lfe 2592000;
        ends 2 2021/05/25 17:46:46;
    }
}

ia-na ">74\342\000\004\357\355\034\270]X\230\032\304\0018sb\000" {
    cltt 0 2021/04/25 17:36:02;
    iaaddr 2001:db8:cafe:1::200 {
        binding state active;
        preferred-life 604800;
        max-lfe 2592000;
        ends 0 2021/05/23 16:56:33;
    }
}

ia-na "\261\360\000\001\000\001\020\1360\010\000\0\364" {
    cltt 5 2021/04/23 18:00:55;
    iaaddr 2001:db8:cafe:1::106 {
        binding state active;
        preferred-life 604800;
        max-lfe 2592000;
    }
}

```

Obrázek 4.5: Databáze leases serveru

Ověření funkčnosti konfigurace SLAAC a stateless DHCPv6 na 1. klientovi:

Na obrázku č. 4.6 lze vidět výpis *ip address* obsahující dvě globální IPv6 adresy vytvořené a přidělené SLAACem pomocí prefixu obsaženém v RA, který rozesílá server pomocí RADVD. Na základě přiřazené adresy je vidět, že se nejedná o adresu vytvořenou pomocí modifikovaného EUI-64, díky chybějícímu *ff-fe* v druhé polovině adresy. Také lze vidět, že klient po žádosti obdržel i DNS servery a doménová jména. Obrázek č. 4.7 je zde jako důkaz, že se jedná o stateless DHCPv6, proto si tedy nevytvořil žádný záznam v databázi leases.

```
Activities □ Terminal dub 25 19:53 cs ▾ 🔍 🗄️ 🌐 📶 🖱️ 🗑️
🔍 🗄️ 🌐 📶 🖱️ 🗑️
valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:fb:47:09 brd ff:ff:ff:ff:ff:ff
    inet fe80::736d:53a2:ccb6:9cf0/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
ubuntu-sles:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:fb:47:09 brd ff:ff:ff:ff:ff:ff
    inet6 2001:d8:cafe:2:ebd2:adb6:d277:8c69/64 scope global temporary dynamic
        valid_lft 86392sec preferred_lft 14392sec
    inet6 2001:d8:cafe:2:32d7:70ed:132c:6ab0/64 scope global dynamic mngntnaddr noprefixroute
        valid_lft 86392sec preferred_lft 14392sec
    inet6 fe80::736d:53a2:ccb6:9cf0/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
ubuntu-sles:~$ sudo cat /run/systemd/resolve/resolv.conf
[sudo] password for ubu:
# This file is managed by man:systemd-resolved(8). Do not edit.
#
# This is a dynamic resolv.conf file for connecting local clients directly to
# all known uplink DNS servers. This file lists all configured search domains.
#
# Third party programs must not access this file directly, but only through the
# symlink at /etc/resolv.conf. To manage man:resolv.conf(5) in a different way,
# replace this symlink by a static file or a different symlink.
#
# See man:systemd-resolved.service(8) for details about the supported modes of
# operation for /etc/resolv.conf.

nameserver 2001:4860:4860::8888
nameserver 2001:4860:4860::8844
search dns.dzi0027.cz dzi0027.cz
ubuntu-sles:~$
```

Obrázek 4.6: Výpis IP adres, DNS serverů a doménových jmen stateless klienta.

Activities Terminal dub 25 19:56 cs

ubu@u-sles: -

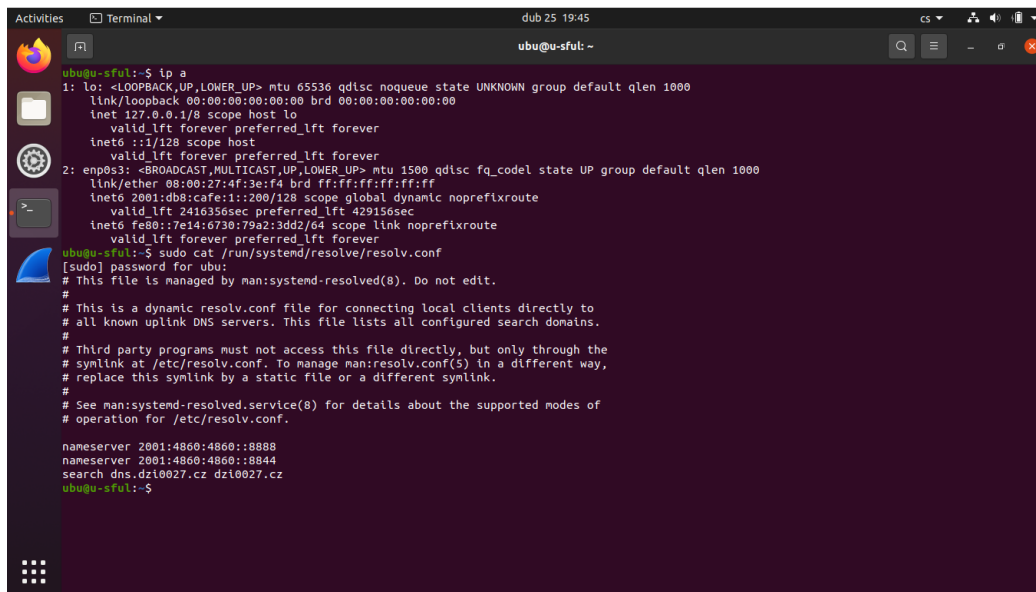
GNU nano 4.8 /var/lib/dhcp/dhclient6.leases

Get Help Write Out Where Is Cut Text Read 8 lines Justify Cur Pos Undo Mark Text To Bracket
 Exit Read File Replace Paste Text To Spell Go To Line Redo Copy Text Where Was

Obrázek 4.7: Databáze leases stateless klienta

Ověření funkčnosti konfigurace statefull DHCPv6 na 2. klientovi:

Obrázek č. 4.8 je důkazem správného přiřazení IPv6 adresy z adresního prostoru 2001:db8:cafe:1::100 - 2001:db8:cafe:1::200. IPv6 adresu i DNS servery a doménová jména obdržel pomocí ISC DHCPv6 serveru. Obrázek č. 4.9 obsahuje všechny klientovy záznamy databáze leases. Záznamy databáze leases mají odlišný formát zápisu od serveru.



```
ubu@u-sful:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:4f:3e:f4 brd ff:ff:ff:ff:ff:ff
    inet6 2001:db8:cafe:1::200/128 scope global dynamic noprefixroute
        valid_lft 2416356sec preferred_lft 429156sec
    inet6 fe80::214:6730:7922:3dd2/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
ubu@u-sful:~$ sudo cat /run/systemd/resolve/resolv.conf
[sudo] password for ubu:
# This file is managed by man:systemd-resolved(8). Do not edit.
#
# This is a dynamic resolv.conf file for connecting local clients directly to
# all known upstream DNS servers. This file lists all configured search domains.
#
# Third party programs must not access this file directly, but only through the
# symlink at /etc/resolv.conf. To manage man:resolved.conf(5) in a different way,
# replace this symlink by a static file or a different symlink.
#
# See man:systemd-resolved.service(8) for details about the supported modes of
# operation for /etc/resolv.conf.
nameserver 2001:4860:4860::8888
nameserver 2001:4860:4860::8844
search dns.dzi0027.cz dzi0027.cz
ubu@u-sful:~$
```

Obrázek 4.8: Výpis ip address, DNS serverů a doménových jmen statefull klienta

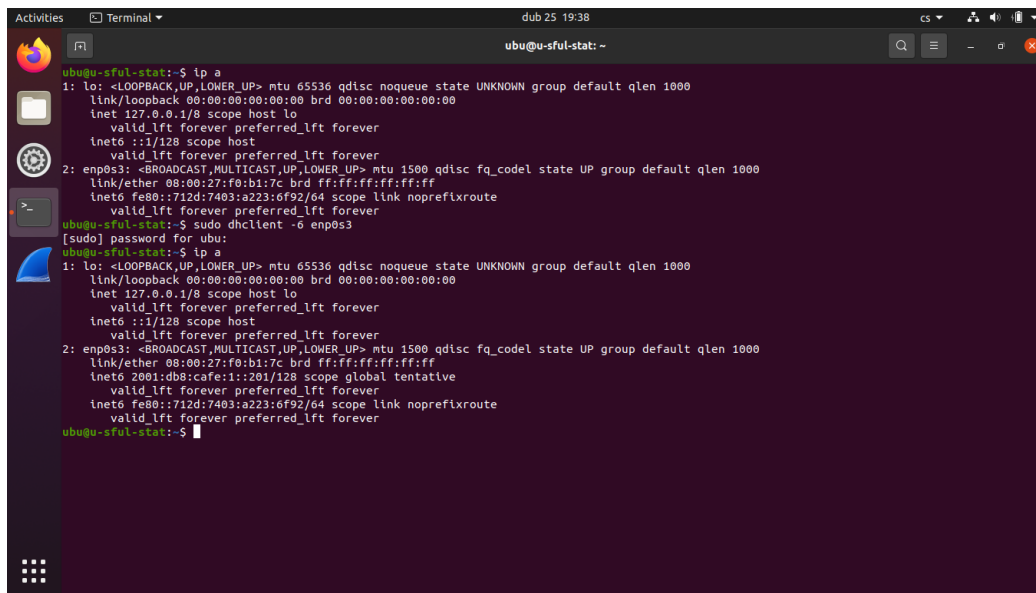


```
CNU nano 4.8
/var/lib/dhcp/dhclient6.leases
default-duid "\000\001\000\001\025\255k\010\000\0>\364";
lease6 {
    interface "enp0s3";
    ia-na 27:4f:3e:f4 {
        starts 1619372805;
        renew 3600;
        rebind 7200;
        iaaddr 2001:db8:cafe:1::1e0 {
            starts 1619372805;
            preferred-life 604800;
            max-life 2592000;
        }
    }
    released;
    option dhcp6.client-id 0:1:0:1:28:15:ad:6b:8:0:27:4f:3e:f4;
    option dhcp6.server-id 0:1:0:1:28:15:b8:7f:8:0:27:1f:37:ec;
    option dhcp6.name-servers 2001:4860:4860::8888,2001:4860:4860::8844;
    option dhcp6.domain-search "dns.dzi0027.cz.", "dzi0027.cz.";
}
lease6 {
    interface "enp0s3";
    ia-na 27:4f:3e:f4 {
        starts 1619372982;
        renew 3600;
        rebind 7200;
        iaaddr 2001:db8:cafe:1::1e0 {
            starts 1619372982;
            preferred-life 604800;
            max-life 2592000;
        }
    }
    option dhcp6.client-id 0:1:0:1:28:15:ad:6b:8:0:27:4f:3e:f4;
    option dhcp6.server-id 0:1:0:1:28:15:b8:7f:8:0:27:1f:37:ec;
    option dhcp6.name-servers 2001:4860:4860::8888,2001:4860:4860::8844;
```

Obrázek 4.9: Databáze leases statefull klienta

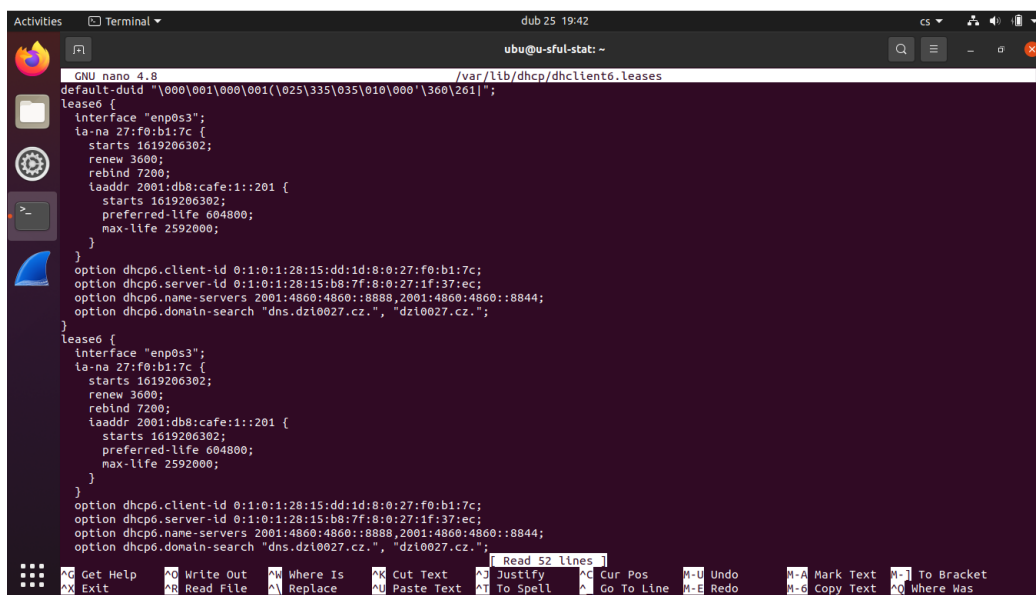
Ověření funkčnosti přiřazení fixní adresy na 3. klientovi:

Na obrázku č. 4.10 lze vidět výpis příkazu *ip address* těsně po zapnutí zařízení, kdy ještě není přiřazena žádná IPv6 adresa. Po žádosti o přidělení adresy pro zvolené rozhraní *sudo dhclient -6 enp0s3* již výpis obsahuje přidělenou fixní adresu 2001:db8:cafe:1::201, která se nachází za adresním prostorem DHCPv6 serveru. Protože je adresa přidělena statefull DHCPv6 serverem, tak databáze leases na obrázku č. 4.11 obsahuje záznam, stejně jako předchozí klient.



```
ubu@u-sful-stat:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:f0:b1:7c brd ff:ff:ff:ff:ff:ff
    inet6 fe80::712d:7403:a223:6f92/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
ubu@u-sful-stat:~$ sudo dhclient -6 enp0s3
[sudo] password for ubu:
ubu@u-sful-stat:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:f0:b1:7c brd ff:ff:ff:ff:ff:ff
    inet6 2001:db8:cafe:1::201/128 scope global tentative
        valid_lft forever preferred_lft forever
    inet6 fe80::712d:7403:a223:6f92/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
ubu@u-sful-stat:~$
```

Obrázek 4.10: Výpis ip address, DNS serverů a doménových jmen statefull klienta s fixní adresou



```
GNU nano 4.8 /var/lib/dhcp/dhclient6.leases
default:duid "\000\001\000\001(\025\335\035\010\000\360\261)";
lease6 {
    interface "enp0s3";
    ia-na 27:f0:b1:7c {
        starts 1619206302;
        renew 3600;
        rebind 7200;
        iaaddr 2001:db8:cafe:1::201 {
            starts 1619206302;
            preferred-life 604800;
            max-life 2592000;
        }
    }
    option dhcp6.client-id 0:1:0:1:28:15:dd:1d:8:0:27:f0:b1:7c;
    option dhcp6.server-id 0:1:0:1:28:15:b8:7f:8:0:27:1f:37:ec;
    option dhcp6.name-servers 2001:4860:4860::8888,2001:4860:4860::8844;
    option dhcp6.domain-search "dns.dzi0027.cz.", "dzi0027.cz.";
}
lease6 {
    interface "enp0s3";
    ia-na 27:f0:b1:7c {
        starts 1619206302;
        renew 3600;
        rebind 7200;
        iaaddr 2001:db8:cafe:1::201 {
            starts 1619206302;
            preferred-life 604800;
            max-life 2592000;
        }
    }
    option dhcp6.client-id 0:1:0:1:28:15:dd:1d:8:0:27:f0:b1:7c;
    option dhcp6.server-id 0:1:0:1:28:15:b8:7f:8:0:27:1f:37:ec;
    option dhcp6.name-servers 2001:4860:4860::8888,2001:4860:4860::8844;
    option dhcp6.domain-search "dns.dzi0027.cz.", "dzi0027.cz.";
}
```

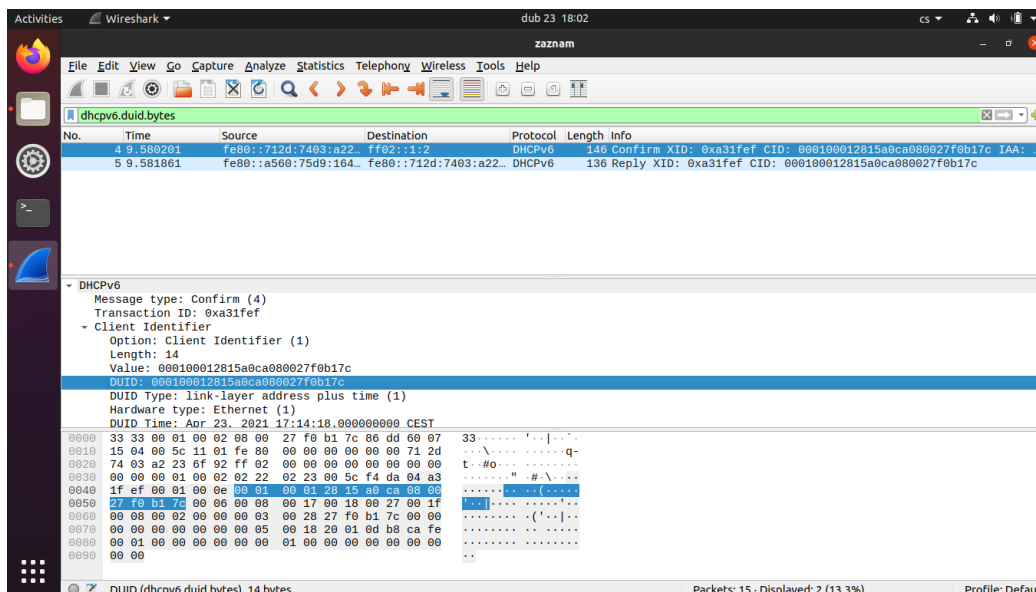
Obrázek 4.11: Databáze leases statefull klienta s fixní adresou

Obrázek č. 4.12 je výpis ostatních konfiguračních parametrů. Na obrázku č. 4.13 lze vidět okno Wiresharku vyfiltrované DHCPv6 komunikace díky `sudo tcpdump -i enp0s3 -w zaznam`, kde je ve zprávě reply vidět DUID. Defaultní DUID lze také získat z databáze klienta leases, ale tam je uvedeno v hexadecimálním tvaru, proto je zapotřebí vytvořit skript pro převedení. DUID využívám jako client-id, které jsem zadal pouze do konfiguračního souboru serveru, ale bohužel toto řešení bylo nefunkční a server tohoto klienta nerozeznal od ostatních. K funkci přidělování fixní adresy bylo potřeba v konfiguračním souboru klienta definovat, aby odesílal zadaný client-id. Pro upřesnění jsem definoval i MAC adresu klienta v konfiguračním souboru serveru.

```
ubu@u-sful-stat:~$ cat /run/systemd/resolve/resolv.conf
# This file is managed by man:systemd-resolved(8). Do not edit.
#
# This is a dynamic resolv.conf file for connecting local clients directly to
# all known uplink DNS servers. This file lists all configured search domains.
#
# Third party programs must not access this file directly, but only through the
# symlink at /etc/resolv.conf. To manage man:resolv.conf(5) in a different way,
# replace this symlink by a static file or a different symlink.
#
# See man:systemd-resolved.service(8) for details about the supported modes of
# operation for /etc/resolv.conf.

nameserver 2001:4860:4860::8888
nameserver 2001:4860:4860::8844
search dns.dzi0027.cz dzi0027.cz
ubu@u-sful-stat:~$
```

Obrázek 4.12: Databáze leases statefull klienta s fixní adresou



Obrázek 4.13: Zachycení DUID klienta pomocí tcpdump, zobrazené ve Wiresharku

Záznamy komunikace ve Wiresharku:

Úryvek zachycené komunikace serveru s 1. klientem na obrázku č. 4.14 zobrazuje 2 typy DHCPv6. Zpráva information request symbolizuje bezstavovou komunikaci, kde klient žádá pouze o dodatečné informace, které jsou konkrétně zobrazeny v dolní části obrázku. Na obrázku č. 4.15 se nachází kompletní DHCPv6 komunikace 2. klienta se serverem. Komunikace obsahuje i uvolnění IPv6 adresy a jeho potvrzení.

160	272.781944	fe80::736d:53a2:ccb...ff02::1:2	DHCPv6	98	Information-request XID: 0xf5aa1 CID: 00030001080027fb4709
161	272.782129	fe80::a560:75d9:164...fe80::736d:53a2:ccb...	DHCPv6	166	Reply XID: 0xf5aa1 CID: 00030001080027fb4709

```

- Client Identifier
  Option: Client Identifier (1)
  Length: 10
  Value: 00030001080027fb4709
  DUID: 00030001080027fb4709
  DUID Type: Link-layer address (3)
  Hardware type: Ethernet (1)
  Link-layer address: 08:00:27:fb:47:09
- Option Request
  Option: Option Request (6)
  Length: 8
  Value: 001700180027001f
  Requested Option code: DNS recursive name server (23)
  Requested Option code: Domain Search List (24)
  Requested Option code: Fully Qualified Domain Name (39)
  Requested Option code: Simple Network Time Protocol Server (31)

```

Obrázek 4.14: Stateless komunikace DHCPv6 serveru s 1. klientem

No.	Time	Source	Destination	Protocol	Length Info
2	26.527169	fe80::7e14:6730:79a2:3dd2	ff02::1:2	DHCPv6	164 Release XID: 0x5271aa CID: 000100012815ad6b0800274f3ef4 IAA
3	26.528768	fe80::a560:7d5d:1c49:d405	fe80::7e14:6730:79a2:3dd2	DHCPv6	125 Reply XID: 0x5271aa CID: 000100012815ad6b0800274f3ef4
4	26.529423	fe80::7e14:6730:79a2:3dd2	ff02::1:2	DHCPv6	148 Solicit XID: 0xa1d0cc IAA: 000100012815ad6b0800274f3ef4 IAA
7	27.15927758	fe80::a560:7d5d:1c49:d405	fe80::7e14:6730:79a2:3dd2	DHCPv6	214 Advertise XID: 0xa1d0cc IAA: 2001:db8:cafe:1::1e0 CID: 000100012815ad6b0800274f3ef4 IAA
8	28.100297803	fe80::7e14:6730:79a2:3dd2	ff02::1:2	DHCPv6	164 Request XID: 0x554999 CID: 000100012815ad6b0800274f3ef4 IAA
L	29.100299074	fe80::a560:7d5d:1c49:d405	fe80::7e14:6730:79a2:3dd2	DHCPv6	214 Reply XID: 0x554999 IAA: 2001:db8:cafe:1::1e0 CID: 000100012815ad6b0800274f3ef4 IAA


```

- IA Address
  Option: IA Address (5)
  Length: 24
  Value: 20010db8cafe000100000000000001e00093a8000278d00
  IPv6 address: 2001:db8:cafe:1::1e0
  Preferred lifetime: 604800
  Valid lifetime: 2592000

- Client Identifier
  Option: Client Identifier (1)
  Length: 14
  Value: 000100012815ad6b0800274f3ef4
  DUID: 000100012815ad6b0800274f3ef4
  DUID Type: link-layer address plus time (1)
  Hardware type: Ethernet (1)
  DUID Time: Apr 23, 2021 18:08:11.000000000 CEST
  Link-Layer address: 08:00:27:4f:3e:f4

- Server Identifier
  Option: Server Identifier (2)
  Length: 14
  Value: 00010001281992e50800271f37ec
  DUID: 00010001281992e50800271f37ec
  DUID Type: link-layer address plus time (1)
  Hardware type: Ethernet (1)
  DUID Time: Apr 26, 2021 17:04:05.000000000 CEST
  Link-Layer address: 08:00:27:1f:37:ec

- DNS recursive name server
  Option: DNS recursive name server (23)
  Length: 32
  Value: 20014860486000000000000000000882001486048600000...
    1 DNS server address: 2001:4860:4860::8888
    2 DNS server address: 2001:4860:4860::8844

- Domain Search List
  Option: Domain Search List (24)
  Length: 28
  Value: 03646e7307647a693030323702637a0007647a6930303237..
- DNS Domain Search List
  Domain Search List FQDN: dns.dzi0027.cz
  Domain Search List FQDN: dzi0027.cz

```

Obrázek 4.15: Statefull komunikace DHCPv6 serveru s 2. klientem

Kapitola 5

Závěr

Tato bakalářská práce obsahuje popis postupu při vytváření malé lokální sítě v rámci simulačního síťového prostředí GNS3. V topologi jsou využita 4 virtuální zařízení běžící na operačním systému Ubuntu Linux zprovozněné ve virtualizačním prostředí Oracle VM VirtualBox. Zařízení jsou propojena pomocí switchů neboli přepínačů. Jedno z VM (virtuálních zařízení) slouží jako DHCPv6 server, další 3 VM slouží jako klienti, kterým budou přiřazeny adresy a další konfigurační parametry, například DNS server, jinými způsoby. První z klientů si vytvoří adresu na základě router advertisementu pomocí metody SLAAC a další konfigurační parametry získá pomocí stateless DHCPv6 serveru, o kterém se dozví díky hodnotě 1 O-bitu obsaženém v RA. Druhý klient bude mít adresu přiřazenou přímo statefull DHCPv6 serverem z vyhrazeného adresního prostoru, klient se o této možnosti dozví pomocí hodnoty 1 M-bitu v RA odeslaného serverem. Třetí klient bude mít také adresu přiřazenou statefull DHCPv6 serverem, ale bude se jednat o fixní adresu mimo adresní prostor. Server klienta rozezná na základě klientského identifikátoru zvaného DUID. Statefull DHCPv6 je zprostředkován pomocí ISC DHCP server daemonu a statefull DHCPv6 je realizováno pomocí RADVD daemonu, který slouží k odesílání router advertisementů.

Část závěru bych chtěl věnovat porovnání teoretické části s praktickou. Na obrázku 4.14 lze vidět vyfiltrovaný úryvek komunikace 1. klienta se serverem. Komunikace byla zachycena pomocí tcpdump a zobrazena v programu Wireshark. Jedná se o DHCPv6 zprávu information request. V těle zprávy lze vidět link-layer adresu, která se shoduje s MAC adresou klienta, který získal globální unicastovou adresu pomocí SLAAC. V sekci option request lze vidět kódy jednotlivých možností jako je DNS rekurzivní server atd. Zpráva reply obsahuje konfigurační parametry, o které si klient zažádal. Úryvek je totožný s průběhem komunikace, který jsem popisoval v teoretické části 2.13.

Na druhém obrázku lze vidět celou DHCPv6 komunikaci 2. klienta se serverem. Komunikace začíná uvolněním adresy a konfiguračních parametrů na mou žádost pomocí příkazu `sudo dhclient -6 -r enp0s3`. Server na uvolnění zprávy zareagoval aktualizací databáze leases, kde vymazal záznam a uvolnil tak adresu pro ostatní klienty. Následuje odpověď serveru s potvrzením. Poté jsem vypnul a zapnul rozhraní a požádal manuálně o adresu, čímž byla vyslána žádost solicit. Server odpověděl

klientovi zprávou advertise, čímž říká klientovi, že je k dispozici služba statefull DHCPv6 serveru. Na základě toho klient pošle zprávu request požadující IPv6 adresu a další konfigurační parametry, které byly nastaveny v konfiguračním souboru klienta. DHCPv6 server odpoví zprávou reply obsahující IPv6 adresu přiřazenou z adresního prostoru serveru a další informace. DHCPv6 komunikace se opět shoduje s komunikací popsanou v teoretické části 2.12, konkrétně na obrázku č. 2.13. V části destination lze také vidět, že klient jako cílovou adresu používá multicastovou adresu ff02::1:2, která je určená všem DHCP relay-agentům a serverům, viz 2.5.

V praktické části popisuji postup konfigurace výše popsaných zařízení. Při konfiguraci jsem se setkal s několika problémy, které se mi zdárně podařilo vyřešit. Jednalo se například o problém při rozeznávání 3. klienta. Klient nebyl rozeznán a dostával tedy IPv6 adresu z adresního poolu namísto fixně vyhrazené adresy. Funkčnost praktické části je ověřena v její poslední sekci pomocí popsaných screenshotů provedených na virtuálních zařízeních.

Přínos své práce vidím v jedinečnosti, protože při hledání návodu, nebo pomocných materiálů jsem se nesetkal s řešením tohoto konkrétního zapojení. Většina návodů, které jsem našel řešily ISC DHCP v reálném prostředí. Pokud jsem našel řešení ISC DHCP v prostředí GNS3, tak se jednalo o DHCP pro IPv4, nebo nebyly přiloženy konfigurační soubory ani jiný popis provedených změn. Většina řešení DHCPv6 byla prováděna právě prostřednictvím routerů a ne na Ubuntu Desktop, takže mi při řešení nebyly přínosem.

Při rozšiřování své práce bych využil cloudu a webového prostředí, čímž bych umožnil přístup ke své práci z jiných zařízení. Dále bych nahradil Ubuntu Desktop využitého jako server za Linuxový Docker Container, který izoluje poskytované služby, takže jedna služba do druhé nezasahuje. Také bych využil novějšího DHCP daemona Kea, který poskytuje mnohem více možností konfigurace. Kea je software od stejných vývojářů jako ISC DHCP. Vývojáři doporučují migrovat z ISC DHCP na údajně čistější a jednodušší Kea [30]. Mě však odradil dlouhý defaultní konfigurační soubor poskytující nespočet možností, také jsem se o Kea dozvěděl pozdě, až při konfiguraci fixní adresy 3. klienta.

Prostředí GNS3 bylo přehledné a proto se mi v něm dobře pracovalo, ale chvíli mi trvalo, než jsem ho zprovoznil ke svým potřebám pro splnění této práce, proto jsem tumuto problému věnoval část své práce. Prostředí lze srovnat například s programem Cisco Packet Tracer, které lze také využít k přípravě na různé certifikační zkoušky. U GNS3 vidím jednu nevýhodu a tou je přenositelnost projektů mezi uživateli. Pokud bych poslal jinému uživateli svůj projekt, je zapotřebí, aby měl nainstalované stejné VM s totožným pojmenováním, jaké jsem používal já. Osobně si však myslím, že GNS3 má větší potenciál pro praktické využití než CPT, díky možnosti využití IOSů reálných zařízení, či IOSů dostupných na internetu.

Literatura

1. R. HINDEN, S. DEERING. *IP Version 6 Addressing Architecture* [Internet Requests for Comments]. RFC Editor, 2006-02. RFC, 4291. RFC Editor. Dostupné také z: <https://tools.ietf.org/html/rfc4291>.
2. S. KAWAMURA, M. KAWASHIMA. *A Recommendation for IPv6 Address Text Representation* [Internet Requests for Comments]. RFC Editor, 2010-08. RFC, 5952. RFC Editor. ISSN 2070-1721. Dostupné také z: <https://tools.ietf.org/html/rfc5952>.
3. T. BERNERS-LEE, R. FIELDING, L. MASINTER. *Uniform Resource Identifier (URI): Generic Syntax* [Internet Requests for Comments]. RFC Editor, 2005-08. RFC, 3986. RFC Editor. Dostupné také z: <https://tools.ietf.org/html/rfc3986>.
4. SATRAPA, Pavel. *IPv6*. Milešovská 5, 130 00 Praha 3: CZ.NIC, 2019. ISBN 978-80-88168-46-1.
5. *IANA IPv6 Special-Purpose Address Registry Created*. 2019-09. Tech. zpr. IANA. Dostupné také z: <https://www.iana.org/assignments/iana-ipv6-special-registry/iana-ipv6-special-registry.xhtml>.
6. *Internet Protocol Version 6 Address Space*. 2019-09. Tech. zpr. IANA. Dostupné také z: <https://www.iana.org/assignments/ipv6-address-space/ipv6-address-space.xhtml>.
7. R. HINDEN, B. HABERMAN. *Unique Local IPv6 Unicast Addresses* [Internet Requests for Comments]. RFC Editor, 2005-10. RFC, 4193. RFC Editor. Dostupné také z: <https://tools.ietf.org/html/rfc4193>.
8. T. MRUGALSKI, M. SIODELSKI, B. VOLZ, A. YOURTCHENKO, M. RICHARDSON, S. JIANG, T. LEMON, T. WINTERS. *Dynamic Host Configuration Protocol for IPv6 (DHCPv6)* [Internet Requests for Comments]. RFC Editor, 2018-11. RFC, 8415. RFC Editor. ISSN 2070-1721. Dostupné také z: <https://tools.ietf.org/pdf/rfc8415.pdf>.
9. R. HINDEN, S. DEERING, E. NORDMARK. *IPv6 Global Unicast Address Format* [Internet Requests for Comments]. RFC Editor, 2003-08. RFC, 3587. RFC Editor. Dostupné také z: <https://tools.ietf.org/pdf/rfc3587.pdf>.

10. B. CARPENTER, S. JIANG. *Significance of IPv6 Interface Identifiers* [Internet Requests for Comments]. RFC Editor, 2014-02. RFC, 7136. RFC Editor. ISSN 2070-1721. Dostupné také z: <https://tools.ietf.org/html/rfc7136>.
11. DAVIES, Joseph. *The Cable Guy IPv6 Autoconfiguration in Windows Vista* [online]. 2016 [cit. 2016-09-07]. Dostupné z: [https://docs.microsoft.com/en-us/previous-versions/technet-magazine/cc137983\(v=msdn.10\)?redirectedfrom=MSDN](https://docs.microsoft.com/en-us/previous-versions/technet-magazine/cc137983(v=msdn.10)?redirectedfrom=MSDN).
12. T. NARTEN, R. DRAVES, S. KRISHNAN. *Privacy Extensions for Stateless Address Autoconfiguration in IPv6* [Internet Requests for Comments]. RFC Editor, 2007-09. RFC, 4941. RFC Editor. Dostupné také z: <https://tools.ietf.org/html/rfc4941>.
13. COFFEEN, Tom. *IPv6 Back to Basics: Interface IDs* [online]. 2018 [cit. 2018-12-12]. Dostupné z: <https://blogs.infoblox.com/ipv6-coe/ipv6-back-to-basics-interface-ids/>.
14. WIKIPEDIA CONTRIBUTORS. *Zero-configuration networking*. 2021. Dostupné také z: https://en.wikipedia.org/wiki/Zero-configuration_networking. [Online; accessed 10-April-2021].
15. S. CHESHIRE, M. KROCHMAL. *Multicast DNS* [Internet Requests for Comments]. RFC Editor, 2013-02. RFC, 6762. RFC Editor. ISSN 2070-1721. Dostupné také z: <https://tools.ietf.org/pdf/rfc6762.pdf>.
16. S. CHESHIRE, M. KROCHMAL. *DNS-Based Service Discovery* [Internet Requests for Comments]. RFC Editor, 2013-02. RFC, 6763. RFC Editor. ISSN 2070-1721. Dostupné také z: <https://tools.ietf.org/pdf/rfc6763.pdf>.
17. DANIEL H STEINBERG, Stuart Cheshire. *Zero Configuration Networking: The Definitive Guide: The Definitive Guide*. oreilly.com, 2005. ISBN 978-0-596-10100-8.
18. MAJOROS, Bc. Lukáš. *Sdílení souborů s využitím Zeroconf a protokolu Samba* [Bachelor Thesis]. 2020. Dostupné také z: <https://dspace.vsb.cz/handle/10084/140540>.
19. T. NARTEN, E. NORDMARK, W. SIMPSON, H. SOLIMAN. *Neighbor Discovery for IP version 6 (IPv6)* [Internet Requests for Comments]. RFC Editor, 2007-09. RFC, 4861. RFC Editor. Dostupné také z: <https://tools.ietf.org/html/rfc4861>.
20. S. THOMSON, T. NARTEN. *IPv6 Stateless Address Autoconfiguration* [Internet Requests for Comments]. RFC Editor, 1998-12. RFC, 2462. RFC Editor. Dostupné také z: <https://tools.ietf.org/pdf/rfc2462.pdf>.
21. S. THOMSON, T. NARTEN, T. JINMEI. *IPv6 Stateless Address Autoconfiguration* [Internet Requests for Comments]. RFC Editor, 2007-09. RFC, 4862. RFC Editor. Dostupné také z: <https://tools.ietf.org/pdf/rfc4862.pdf>.

22. P. LEACH, M. MEALLING, R. SALZ. *A Universally Unique Identifier (UUID) URN Namespace* [Internet Requests for Comments]. RFC Editor, 2005-07. RFC, 4122. RFC Editor. Dostupné také z: <https://tools.ietf.org/pdf/rfc4122.pdf>.
23. R. DROMS. *Stateless Dynamic Host Configuration Protocol (DHCP) Service for IPv6* [Internet Requests for Comments]. RFC Editor, 2004-04. RFC, 3736. RFC Editor. Dostupné také z: <https://tools.ietf.org/pdf/rfc3736.pdf>.
24. *Getting Started with GNS3*. Dostupné také z: <https://docs.gns3.com/docs/>.
25. *Download GNS3*. Dostupné také z: <https://www.gns3.com/software/download>.
26. *Appliances*. Dostupné také z: <https://www.gns3.com/marketplace/appliances>.
27. *Download GNS3 VM*. Dostupné také z: <https://www.gns3.com/software/download-vm>.
28. *Download Ubuntu Desktop*. Dostupné také z: <https://ubuntu.com/download/desktop>.
29. *radvd.conf(5) - Linux man page*. Dostupné také z: <https://linux.die.net/man/5/radvd.conf>.
30. *Migrating from ISC DHCP to Kea DHCP using the Migration Assistant*. Dostupné také z: <https://kb.isc.org/docs/migrating-from-isc-dhcp-to-kea-dhcp-using-the-migration-assistant>.

```
# Server configuration file example for DHCPv6
# From the file used for TAHI tests - addresses chosen
# to match TAHI rather than example block.

# IPv6 address valid lifetime
# (at the end the address is no longer usable by the client)
# (set to 30 days, the usual IPv6 default)
default-lease-time 2592000;

# IPv6 address preferred lifetime
# (at the end the address is deprecated, i.e., the client should use
# other addresses for new connections)
# (set to 7 days, the usual IPv6 default)
preferred-lifetime 604800;

# T1, the delay before Renew
# (default is 1/2 preferred lifetime)
# (set to 1 hour)
option dhcp-renewal-time 3600;

# T2, the delay before Rebind (if Renews failed)
# (default is 3/4 preferred lifetime)
# (set to 2 hours)
option dhcp-rebinding-time 7200;

# Enable RFC 5007 support (same than for DHCPv4)
allow leasequery;

# Global definitions for name server address(es) and domain search list
option dhcp6.name-servers 2001:4860:4860::8888, 2001:4860:4860::8844;
option dhcp6.domain-search "dns.dzi0027.cz","dzi00227.cz";

# Set preference to 255 (maximum) in order to avoid waiting for
# additional servers when there is only one
##option dhcp6.preference 255;

# Server side command to enable rapid-commit (2 packet exchange)
```

```

##option dhcp6.rapid-commit;

# The delay before information-request refresh
# (minimum is 10 minutes, maximum one day, default is to not refresh)
# (set to 6 hours)
option dhcp6.info-refresh-time 21600;

# Static definition (must be global)
host u_sful_static {
# # The entry is looked up by this
    host-identifier option dhcp6.client-id 00:01:00:01:28:15:a0:ca:08:00:27:f0:b1:7
        c;
    hardware ethernet 08:00:27:f0:b1:7c;
#
# # A fixed address
    fixed-address6 2001:db8:cafe:1::201;
#
# # A fixed prefix
    fixed-prefix6 3ffe:501:ffff:101::/64;
# # Override of the global definitions,
# # works only when a resource (address or prefix) is assigned
# option dhcp6.name-servers 3ffe:501:ffff:100:200:ff:fe00:4f4e;
#
# # For debug (to see when the entry statements are executed)
# # (log "sol" when a matching Solicitation is received)
# ##if packet(0,1) = 1 { log(debug,"sol"); }
}
#
#host otherclient {
#     # This host entry is hopefully matched if the client supplies a DUID-LL
#     # or DUID-LLT containing this MAC address.
#     hardware ethernet 01:00:80:a2:55:67;
#
#     fixed-address6 3ffe:501:ffff:100::4321;
#}
#authoritative;
# The subnet where the server is attached
# (i.e., the server has an address in this subnet)

```

```

subnet6 2001:db8:cafe:1::/64 {
# # Two addresses available to clients
# # (the third client should get NoAddrsAvail)
    range6 2001:db8:cafe:1::100 2001:db8:cafe:1::200;
    option dhcp6.name-servers 2001:4860:4860::8888, 2001:4860:4860::8844;
    option dhcp6.domain-search "dns.dzi0027.cz","dzi0027.cz";
}

# # Use the whole /64 prefix for temporary addresses
# # (i.e., direct application of RFC 4941)
# range6 3ffe:501:ffff:100:: temporary;
#
# # Some /64 prefixes available for Prefix Delegation (RFC 3633)
# prefix6 3ffe:501:ffff:100:: 3ffe:501:ffff:111:: /64;
#}

# A second subnet behind a relay agent
#subnet6 3ffe:501:ffff:101::/64 {
# range6 3ffe:501:ffff:101::10 3ffe:501:ffff:101::11;
#
# # Override of the global definitions,
# # works only when a resource (address or prefix) is assigned
# option dhcp6.name-servers 3ffe:501:ffff:101:200:ff:fe00:3f3e;
#
#}

# A third subnet behind a relay agent chain
#subnet6 3ffe:501:ffff:102::/64 {
# range6 3ffe:501:ffff:102::10 3ffe:501:ffff:102::11;
#}

```

A 1: Konfigurační soubor dhcpd6.conf

```

interface enp0s3
{
AdvSendAdvert on;
AdvManagedFlag on;
AdvOtherConfigFlag on;
MinRtrAdvInterval 30;

```

```
MaxRtrAdvInterval 100;
prefix 2001:db8:cafe:2::/64
{
  AdvOnLink on;
  AdvAutonomous on;
  AdvRouterAddr off;
};

};
```

A 2: Konfigurační soubor radvd.conf

```
# Defaults for isc-dhcp-server (sourced by /etc/init.d/isc-dhcp-server)

# Path to dhcpd's config file (default: /etc/dhcp/dhcpd.conf).
#DHCPDv4_CONF=/etc/dhcp/dhcpd.conf
DHCPDv6_CONF=/etc/dhcp/dhcpd6.conf

# Path to dhcpd's PID file (default: /var/run/dhcpd.pid).
#DHCPDv4_PID=/var/run/dhcpd.pid
DHCPDv6_PID=/var/run/dhcpd6.pid

# Additional options to start dhcpd with.
# Don't use options -cf or -pf here; use DHCPD_CONF/ DHCPD_PID instead
#OPTIONS=""

# On what interfaces should the DHCP server (dhcpd) serve DHCP requests?
# Separate multiple interfaces with spaces, e.g. "eth0 eth1".
INTERFACESv4=""
INTERFACESv6="enp0s3"
```

A 3: Konfigurační soubor pro nastavení rozhraní ISC DHCP

```
# The format of this file is documented in the dhcpd.leases(5) manual page.
# This lease file was written by isc-dhcp-4.4.1

# authoring-byte-order entry is generated, DO NOT DELETE
authoring-byte-order little-endian;
```

```

server-duid "\000\001\000\001(\025\270\177\010\000'\0377\354";

ia-na "\364>0'\000\001\000\001(\025\255k\010\000'0>\364" {
    cltt 0 2021/04/25 17:49:43;
    iaaddr 2001:db8:cafe:1::1e0 {
        binding state active;
        preferred-life 604800;
        max-life 2592000;
        ends 2 2021/05/25 17:49:43;
    }
}

ia-na ">?4\342\000\004\357\355(\034\270]Xx\230\032\304\0018sb\000" {
    cltt 0 2021/04/25 18:47:40;
    iaaddr 2001:db8:cafe:1::200 {
        binding state active;
        preferred-life 604800;
        max-life 2592000;
        ends 0 2021/05/23 16:56:33;
    }
}

ia-na "|\261\360'\000\001\000\001(\020|\360\010\000'0>\364" {
    cltt 5 2021/04/23 18:00:55;
    iaaddr 2001:db8:cafe:1::106 {
        binding state active;
        preferred-life 604800;
        max-life 2592000;
        ends 0 2021/05/23 17:00:22;
    }
}

```

A 4: Databáze leases na serveru

```

# Configuration file for /sbin/dhclient.
#
# This is a sample configuration file for dhclient. See dhclient.conf's
# man page for more information about the syntax of this file
# and a more comprehensive list of the parameters understood by

```



```

# dhclient.
#
# Normally, if the DHCP server provides reasonable information and does
# not leave anything out (like the domain name, for example), then
# few changes must be made to this file, if any.
#

option rfc3442-classless-static-routes code 121 = array of unsigned integer 8;

send host-name = gethostname();
#request subnet-mask, broadcast-address, time-offset, routers,
# domain-name, domain-name-servers, domain-search, host-name,
# dhcp6.name-servers, dhcp6.domain-search, dhcp6.fqdn, dhcp6.sntp-servers,
# netbios-name-servers, netbios-scope, interface-mtu,
# rfc3442-classless-static-routes, ntp-servers;

request subnet-mask, host-name, dhcp6.name-servers,
        dhcp6.domain-search;

#send dhcp-client-identifier 1:0:a0:24:ab:fb:9c;
#send dhcp-lease-time 3600;
#supersede domain-name "fugue.com home.vix.com";
#prepend domain-name-servers 127.0.0.1;
#require subnet-mask, domain-name-servers;
timeout 300;
retry 60;
#reboot 10;
#select-timeout 5;
#initial-interval 2;
#script "/sbin/dhclient-script";
#media "-link0 -link1 -link2", "link0 link1";
#reject 192.33.137.209;

#alias {
# interface "eth0";
# fixed-address 192.5.5.213;
# option subnet-mask 255.255.255.255;
#}

```

```
#lease {
#   interface "eth0";
#   fixed-address 192.33.137.200;
#   medium "link0 link1";
#   option host-name "andare.swiftmedia.com";
#   option subnet-mask 255.255.255.0;
#   option broadcast-address 192.33.137.255;
#   option routers 192.33.137.250;
#   option domain-name-servers 127.0.0.1;
#   renew 2 2000/1/12 00:00:01;
#   rebind 2 2000/1/12 00:00:01;
#   expire 2 2000/1/12 00:00:01;
#}
```

A 5: Konfigurační soubor dhclient6.conf stateless klienta

```
# Configuration file for /sbin/dhclient.
#
# This is a sample configuration file for dhclient. See dhclient.conf's
# man page for more information about the syntax of this file
# and a more comprehensive list of the parameters understood by
# dhclient.
#
# Normally, if the DHCP server provides reasonable information and does
# not leave anything out (like the domain name, for example), then
# few changes must be made to this file, if any.
#

option rfc3442-classless-static-routes code 121 = array of unsigned integer 8;

send host-name = gethostname();
#request subnet-mask, broadcast-address, time-offset, routers,
# domain-name, domain-name-servers, domain-search, host-name,
# dhcp6.name-servers, dhcp6.domain-search, dhcp6.fqdn, dhcp6.sntp-servers,
# netbios-name-servers, netbios-scope, interface-mtu,
# rfc3442-classless-static-routes, ntp-servers;

send host-name = gethostname();
```

```

request subnet-mask, time-offset, host-name,
    dhcp6.name-servers, dhcp6.domain-search,
    dhcp6.fqdn, interface-mtu;

#send dhcp-client-identifier 1:0:a0:24:ab:fb:9c;
#send dhcp-lease-time 3600;
#supersede domain-name "fugue.com home.vix.com";
#prepend domain-name-servers 127.0.0.1;
#require subnet-mask, domain-name-servers;
timeout 300;
retry 20;
#reboot 10;
#select-timeout 5;
#initial-interval 2;
#script "/sbin/dhclient-script";
#media "-link0 -link1 -link2", "link0 link1";
#reject 192.33.137.209;

#alias {
# interface "eth0";
# fixed-address 192.5.5.213;
# option subnet-mask 255.255.255.255;
#}

#lease {
# interface "eth0";
# fixed-address 192.33.137.200;
# medium "link0 link1";
# option host-name "andare.swiftmedia.com";
# option subnet-mask 255.255.255.0;
# option broadcast-address 192.33.137.255;
# option routers 192.33.137.250;
# option domain-name-servers 127.0.0.1;
# renew 2 2000/1/12 00:00:01;
# rebind 2 2000/1/12 00:00:01;
# expire 2 2000/1/12 00:00:01;
#}

```

```
# Configuration file for /sbin/dhclient.
#
# This is a sample configuration file for dhclient. See dhclient.conf's
# man page for more information about the syntax of this file
# and a more comprehensive list of the parameters understood by
# dhclient.
#
# Normally, if the DHCP server provides reasonable information and does
# not leave anything out (like the domain name, for example), then
# few changes must be made to this file, if any.
#

option rfc3442-classless-static-routes code 121 = array of unsigned integer 8;

#send host-name = gethostname();
#request subnet-mask, broadcast-address, time-offset, routers,
# domain-name, domain-name-servers, domain-search, host-name,
# dhcp6.name-servers, dhcp6.domain-search, dhcp6.fqdn, dhcp6.sntp-servers,
# netbios-name-servers, netbios-scope, interface-mtu,
# rfc3442-classless-static-routes, ntp-servers;

send host-name = gethostname();
request subnet-mask, time-offset, host-name,
        dhcp6.name-servers, dhcp6.domain-search,
        dhcp6.fqdn, interface-mtu;

send dhcp-client-identifier 00:01:00:01:28:15:a0:ca:08:00:27:f0:b1:7c;
#send dhcp-lease-time 3600;
#supersede domain-name "fugue.com home.vix.com";
#prepend domain-name-servers 127.0.0.1;
#require subnet-mask, domain-name-servers;
timeout 300;
retry 60;
#reboot 10;
#select-timeout 5;
```

```
#initial-interval 2;
#script "/sbin/dhclient-script";
#media "-link0 -link1 -link2", "link0 link1";
#reject 192.33.137.209;

#alias {
# interface "eth0";
# fixed-address 192.5.5.213;
# option subnet-mask 255.255.255.255;
#}

#lease {
# interface "eth0";
# fixed-address 192.33.137.200;
# medium "link0 link1";
# option host-name "andare.swiftmedia.com";
# option subnet-mask 255.255.255.0;
# option broadcast-address 192.33.137.255;
# option routers 192.33.137.250;
# option domain-name-servers 127.0.0.1;
# renew 2 2000/1/12 00:00:01;
# rebind 2 2000/1/12 00:00:01;
# expire 2 2000/1/12 00:00:01;
#}
```

A 7: Konfigurační soubor dhclient6.conf statelefull klienta s fixní adresou

